DOCUMENT RESUME

ED 038 266 24 RE 002 714

AUTHOR	Leton, Donald A.
TITLE	Computer Program to Convert Word Orthography to Phoneme Equivalents. Final Report.
INSTITUTION	Hawaii Univ., Honolulu. Educational Research and Development Center.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-8-0114
PUB DATE	Nov 69
GRANT	OEG-9-8-0114-0105(057)
NOTE	118p.
EDRS PRICE	EDRS Price MF-$0.50 HC-$6.00
DESCRIPTORS	*Alphabets, Automation, *Computer Programs, *Phonemes, Primary Grades, Reading, Reading Materials

ABSTRACT

A computer program to convert the orthography of English words to phoneme equivalents was developed. For purposes of this study, a grapheme was defined as the minimum unit of orthography to which a segmental phoneme, a unitary phoneme combination, or a nonphoneme would be associated. An informational processing model for automatic reading was then constructed. A conditional association matrix, with 65 rows for grapheme signals and 45 columns for phoneme designates, was hypothesized as an adequate structure to accommodate all of the English grapheme-phoneme associations. Two phases of the association matrix were described: a dynamic learning phase and a skilled state or processing phase. The acquisition of reading skills was simulated by attempting to write a computer program for each grade level in a basic reading series. Programs were written for the preprimer and primer of the Lippincott series. These programs achieved 96 percent accuracy in processing the 190 words in the preprimer and 85 percent accuracy in processing the 770 words in the primer. The types of errors are discussed. The associations and rules for processing the grade-1 readers in the series have also been written. The results of the research demonstrate the feasibility of automated reading. References and an appendix of data are included. (Author/WB)

EDUCATION RESEARCH AND DEVELOPMENT CENTER

COLLEGE OF EDUCATION
UNIVERSITY OF HAWAII
HONOLULU, HAWAII

FINAL REPORT
Project No. 8-0114
Grant No. 9-8-0114-0105(057)

COMPUTER PROGRAM TO CONVERT WORD ORTHOGRAPHY TO PHONEME EQUIVALENTS

Donald A. Leton
University of Hawaii
1776 University Avenue
Honolulu, Hawaii 96822

November 1969

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

FINAL REPORT

Project No. 8-0114
Grant No. 9-8-0114-0105(057)

COMPUTER PROGRAM TO CONVERT WORD ORTHOGRAPHY TO PHONEME EQUIVALENTS

Donald A. Leton

University of Hawaii

Honolulu, Hawaii

November 1969

The research reported herein was performed pursuant to a grant with the Office of Education, U. S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

Contents

List of Tables and Figures

Summary

The purpose of this study was to develop a computer program to convert the orthography of English words to phoneme equivalents. This is regarded as the critical process in reading. Knowledge about the conversion is pertinent to various problems such as the automation of reading for the blind, the acquisition of reading skills, the efficient instruction of beginning readers and the remedial instruction of disabled readers.

The concept of grapheme-phoneme associations as presented by Fries and Gibson is not adequate for this conversion. Each letter of the alphabet and each punctuation mark is ordinarily defined as a grapheme. For purposes of this study a grapheme was redefined as the minimum unit of orthography to which a segmental phoneme, a unitary phoneme-combination, or a non-phoneme would be associated. Examples of the unitary phoneme combinations are the vowel-semivowel nuclei for diphthongs and the /ks/ for <x>.

An information processing model for automatic reading was then constructed. A conditional association matrix, with 65 rows for grapheme signals and 45 columns for phoneme designates, was hypothesized as an adequate structure to accommodate all of the English grapheme-phoneme associations. Two phases for the association matrix were also described -- a dynamic learning phase and a skilled state, or processing phase. The learning phase consists of: 1) receiving the grapheme signal, 2) designating the correct phoneme unit and 3) identifying the conditions in the word environment which determine the association. This is repeated until all of the legitimate conjunctions are formed and all of the determining conditions are identified for a specific word corpus. The processing phase consists of: 1) receiving the grapheme signals, 2) testing for the presence of conditions in the word

environments and 3) detecting the correct phoneme designates.

The acquisition of reading skills was simulated by attempting to write a computer program for each grade level in a basic reading series. The computer program is an algorithm for segmenting word orthography into graphemes and for selecting the segmental phoneme unit (or the nonphoneme) as a preliminary requirement for its representation (or nonrepresentation) in oral language.

Programs were written for the pre-primer and primer of the Lippincott series. These programs achieved 96 percent accuracy in processing the 190 words in the pre-primer, and 85 percent accuracy in processing the 770 words in the primer. The types of errors were discussed.

The associations and rules for processing the 1-1 and 1-2 readers in the series have also been written. These will require about a six-fold increase in the size of the program and work on this phase is continuing. The adequacy of the program was tested through processing three corpora from the Rinsland vocabulary list. Seventy-one percent accuracy was achieved in processing a 320 word corpus of three-letter words. Only 45 and 29% accuracy was obtained on the 1029 four-letter words and 1332 five-letter words in the list.

A statistical analysis of a grapheme-phoneme matrix obtained in a previous study was included in this report. Hierarchical grouping procedures were used to identify the groups of graphemes which tend to be associated with the same phonemes. The communality of graphemes such as <c> <k> <ck> and <q> is based on their association to /k/. The degree of this communality is also quantified. The grouping or factoring of phoneme units is also discussed.

The results of this research demonstrate the feasibility of automated reading. The essentials for automated reading are identified. Further study of programming strategies to improve the efficiency and the sophistication of processing is needed. No immediate practical or applied value is claimed for the output of segmental phoneme chains. The need for 'piggy-back' programs to superimpose the suprasegmental phonemes in sentence reading, and the subsequent need for tests of comprehensibility, are discussed.

Introduction

Associational learning is generally recognized as the central requirement in learning to read. In discussing the nature and development of reading Judd (1927) explicated the process in his statement, "the printed units must be made to coincide with their oral counterparts". The coincidence of the two systems of language, i.e., English orthography with oral English, is accomplished in the successful teaching of reading. Neither educators nor linguists, however, have identified efficient procedures for converting, or for learning how to convert, the graphic system of English into its phonic equivalents.

Gibson and associates (1962) have defined grapheme-phoneme correspondence as the critical unit in the reading process. They regard this unit as a higher-order unit, with graphemes and phonemes representing the primary variables. Two empirical studies using pseudo words support their hypothesis that reading skill is a function of prior knowledge of grapheme-phoneme associations.

Bloomfield and Barnhart (1961) have suggested the use of phoneme analysis for the teaching of reading and Stratemeyer and Smith (1963) have published a series of primers for elementary school children which incorporates a systematic approach to phonemic associations.

There has been a disagreement among authorities in the field of reading as to whether grapheme-phoneme correspondences are learned as discrete units, or whether an entire phoneme chain as acquired in speaking is associated with printed words and phrases. Learning to read may involve several levels of associational learning. As reading matures perhaps only a few critical grapheme-phoneme associations are necessary to elicit the phoneme chains which characterize fluent reading. For the development of word analysis in beginning reading, however, grapheme-phoneme associations may be more pertinent.

Although there are major correspondences between the English alphabet and English phonemes there are also a number of inconsistent, arbitrary, and unsystematic relationships between the two systems. The manner in which these regular and irregular associations are learned, and the manner in which they should be taught for the most efficient acquisition of reading skills, has not as yet been investigated.

The approach by Pitnam (1962) utilizing an augmented alphabet for teaching reading establishes a more direct relationship between the printed symbols and phonemes. The Augmented Roman Alphabet which has subsequently been named the initial teaching alphabet (i.t.a.) is composed of 43 characters, 14 of which are digraphs. This alphabet translates into 24 consonant phonemes and 16 vowels and diphthongs. This approach involves a departure from traditional orthography in order to effect spelling to sound correspondence.

Venezky (1962) in his early work on spelling-to-sound correlation used a computer to count the total occurrences of letters and letter clusters with the letter sounds appearing in the 19,607 words in the Thorndike Senior Century Dictionary. In his more recent work with Weir (1964) computer routines were used to identify spelling-to-phoneme associations. The phoneme notation was Kenyon and Knott's condensation of the International Phonetic Alphabet (1951). Although they were able to identify a higher degree of spelling-to-sound patterning than was generally recognized they also acknowledged that the traditional view of orthography, i.e. as 26 letter graphemes, is not adequate for spelling-to-sound mapping. They suggested the use of morphophoneme correspondences as relational units and the use of 'marker' letters. These procedures may have important value for research and for teaching reading.

Hanna and associates (1964) have studied the spelling variations of the phonemes appearing in a list of 17,000 American-English words. Their effort was directed toward identifying phoneme-grapheme associations for their possible value in the teaching of spelling. In the second phase of their work Rudorf (1964) developed an algorithm for computer spelling. The program correctly identified the orthography for 8346, or 49 percent, of the 17,000 words. The work of Venezky and Rudorf demonstrates the feasibility of systematizing the conversion of one system to the other.

Problem:

Ordinarily a grapheme is defined as the minimum unit of the graphic system and a phoneme is defined as a minimum class of sounds in the phonic system. Using these traditional definitions the correspondence between units, as suggested by Bloomfield and Gibson, does not exist. Each of the 26 letters of the alphabet and the punctuation marks are identified as segmental graphemes. These do not have a direct and consistent relationship to segmental phonemes.

The purpose of this research is to develop a computer program to convert English word orthography to its phoneme equivalents. In order to do so, and to identify the basic graphic-phonic associations which are implicit in successful reading, it is necessary to redefine terms. In a previous paper (Leton 1967) a segmental grapheme was defined as the minimum unit of orthography to which a segmental phoneme, a phoneme combination, or a non-phoneme could be associated. This strategy of parallel definitions for graphemes and phonemes enables a translation between the two systems. Although this is a departure from the traditional definition it retains the concept of grapheme-phoneme associations as recommended by Bloomfield and by Gibson.

The theoretical model for reading presented in this research is a conditional association matrix. The ability to read is defined as the ability to receive and process orthography, and to associate it to its phonic equivalents. One means of testing this model is to simulate the receiving, processing and converting of the printed symbols through a mechanical or computer system.

Methods

Selection of Word Corpora

The Lippincott Basic Reading Series by McCracken and Walcutt (pre-primer, primer and 1-1) were selected for the preliminary corpora. This series was selected because it is a developmental, linguistic reading program with sequentially structured texts.

Input and Grapheme Segmentation

The letters of the alphabet and punctuation marks are pre-specified as legitimate input. The processing of the word orthography includes two sub-processes: grapheme segmentation and phoneme association.

Sixty-six graphemes were identified as the graphic units which generate all of the segmental phonemes. Thirty-eight of these are digraphs, including consonant combinations such as <gh>, <ng>, and <th>, vowel-vowel combinations such as <ee>, <ea>, and vowel-semivowel combinations such as <ey>, <ay>, etc. Twenty-six are unigraphs, i.e. the letters of the alphabet, and two are trigraphs <eau> and <sci>. Punctuation marks were mapped to open juncture /+/ which is a suprasegmental rather than a segmental phoneme.

Phonemic Notation System

A variation of the phoneme system presented by Trager and Smith (1951) was used in this research. The system includes 33 segmental phonemes, 9 phoneme combinations, one juncture phoneme and a non-phoneme symbol for a silent grapheme. The phonemes and their numeric representations for computer output were as follows. The nine simple vowels /i e æ ɨ ə a u o ɔ/ were assigned the numerals 1 through 9; the twenty consonants /p t k b d g č ǰ f θ h v ð s š z ž l m n ŋ w r y/ were assigned numerals 10 through 33; the consonant combination /ks/, as one of the minimum phoneme representations for <x>, was assigned 34; eight vowel-semivowel nuclei /iy ey ay oy uw ow aw/ were assigned numberals 35 through 41; the combinations /yu/ and /wə/, as minimum representations for <u> as in <use>, and for <o> as in <one>, were assigned 42 and 43 respectively; the juncture phoneme /+/ as minimum representation for transition stop, was assigned 49; and the non-phoneme symbol /∅/ was assigned 50.

The Learning Matrix

The learning matrix is therefore a 66 x 45 grapheme-phoneme matrix. It is hypothesized that this is an adequate structure to include all of the grapheme-phoneme associations pertinent to the conversion of English orthography to segmental phonemes. The rows of the matrix are presented in the following table.

Rows in the Grapheme-Phoneme Matrix

<b> → /b/, /∅/

<c> → /k/, /s/, /č/, /∅/

<ce> → /š/

<ch> → /k/, /č/, /š/

<ci> → /š/

<ck> → /k/

<d> → /d/, /∅/

<f> → /f/, /v/, /∅/

<g> → /g/, /ǰ/, /∅/

<gh> → /g/, /f/, /∅/

<h> → /h/, /∅/

<j> → /ǰ/

<k> → /k/, /∅/

<l> → /l/, /∅/

<m> → /m/, /∅/

<n> → /n/, /ŋ/, /∅/

<ng> → /ŋ/

<p> → /p/, /∅/

<ph> → /f/, /p/

<q> → /k/

<r> → /r/, /∅/

<rh> → /r/

<s> → /s/, /z/, /š/, /∅/

<sci> → /š/

<sh> → /š/

<si> → /š/

<t> → /t/, /ø/

<ti> → /š/, /č/

<th> → /ð/, /θ/

<v> → /v/

<w> → /w/, /ø/

<wh> → /w/, /h/

<x> → /ks/, /z/

<y> → /y/, /i/, /ay/, /iy/

<z> → /z/, /ž/, /ø/

<a> → /e/, /æ/, /ə/, /a/, /ɔ/, /ey/

<e> → /i/, /e/, /ə/, /iy/, /ø/

<i> → /i/, /ə/, /iy/, /ay/, /ø/

<ia> → /ay/

<o> → /ə/, /a/, /u/, /ɔ/, /uw/, /ow/, /wə/

<u> → /w/, /ə/, /u/, /ø/, /yuw/, /uw/

<ie> → /iy/, /ay/, /e/, /i/

<ei> → /i/, /iy/, /ey/, /e/

<ee> → /iy/, /i/

<ea> → /e/, /iy/, /i/, /a/, /ey/

<eau> → /yuw/

<eo> → /iy/

<eu> → /uw/

<ew> → /uw/, /yuw/

<ey> → /iy/, /ey/, /ay/

<ai> → /e/, /æ/, /ey/, /ə/, /ay/

<ay> → /ey/, /ay/

<au> → /ɔ/, /æ/

<aw> → /ɔ/

<ae> → /ay/

<oa> → /ow/, /ɔ/

<oy> → /oy/

<ou> → /ɔ/, /uw/, /ow/, /u/, /ə/, /aw/

<ow> → /ow/, /aw/

<oo> → /uw/, /ə/, /ɔ/, /u/

<oe> → /ow/, /ə/

<oi> → /oy/

<ui> → /i/, /uw/

<uy> → /ay/

<ue> → /uw/, /yuw/

<ye> → /ay/

<.> <:> <,> <-> → /+/

The juncture phoneme /+/ was associated to punctuation marks and word boundary. It is a suprasegmental phoneme included in the matrix to facilitate word processing.

The use of an association matrix was previously illustrated in charting the 23,210 grapheme-phoneme associations which occurred in the 6949 words in the pre-primer and primer of the Ginn Basic Reading series.

The grapheme-phoneme matrix is a row matrix. Each row includes the antecedent grapheme and each of the consequent phonemes which may be designated as output. To illustrate the conditional nature of grapheme-phoneme associations the <gh> grapheme may be designated as /f/, /g/ or /Ø/ as in <tough>, <ghost>, or <dough>. The <ea> in <lead> may be designated as /e/ or /iy/. The phoneme choices indicated above for <gh> are conditional to the word environment whereas the choice for <ea> would require a semantic analysis of the text. The latter is beyond the scope of this project however.

Computer Simulation of Reading

Computer simulation of reading is a double simulation in that the letter characters are simulated in punched cards and subsequently as electrical current, and the phonemes are simulated in the numeric output.

Psychological theory of learning refers to conditioned associations. The simulation of these associations in computer learning are simple on-off circuits for which the logical rules of operation are changed as a function of input information. The learning matrix which represents both the computer circuit and the grapheme-phoneme associations is a zero-one matrix. A portion of this matrix is presented in Figure 1. The ones in each row represent

(Figure 1)

GRAPHEME INPUT	PHONEME OUTPUT								
	/b/	/s/	/k/	/č/	/š/	/d/	/f/	/v/	/ø/
⟨b⟩	1	0	0	0	0	0	0	0	1
⟨c⟩	0	1	1	1	0	0	0	0	1
⟨ch⟩	0	0	1	1	1	0	0	0	0
⟨ck⟩	0	0	1	0	0	0	0	0	0
⟨d⟩	0	0	0	0	0	1	0	0	1
⟨f⟩	0	0	0	0	0	0	1	1	0

FIGURE 1. SECTION OF GRAPHEME – PHONEME LEARNING MATRIX

the set of legitimate phoneme designates for the antecedent grapheme. The zeros are not true zeros, but rather they represent cells in which associations are not legitimate. Successful conversion of orthography to phonemes hinges on operations in this matrix.[1]

The learning phase in reading simulation and the computer processing phase are illustrated in Figure 2. The intersects of these circuits

(Figure 2)

correspond to the cells in the learning matrix. The circuit rows carry signal information and the columns are conductors of the phoneme designates. The zero-one matrix has value for studying the logical operation of the associations. The occurrence or non-occurrence of the expected phoneme output can be related to rules of association.

The learning phase is a dynamic phase during which the associations are initiated and the conditions or 'rules' for their formation are accumulated. In words of the learning psychologist the phoneme responses which are already available in oral language are brought under control of the visual stimuli in the orthography.

The second phase is a stable phase during which the system reads. In the machine system this is referred to as the processing phase and in human learning this is recognized as the skilled state, or reading achievement.

---

1 A reversal of the input-output information would produce the type of matrix pertinent to spelling, stenography, or other graphic representation of a phonic stream.

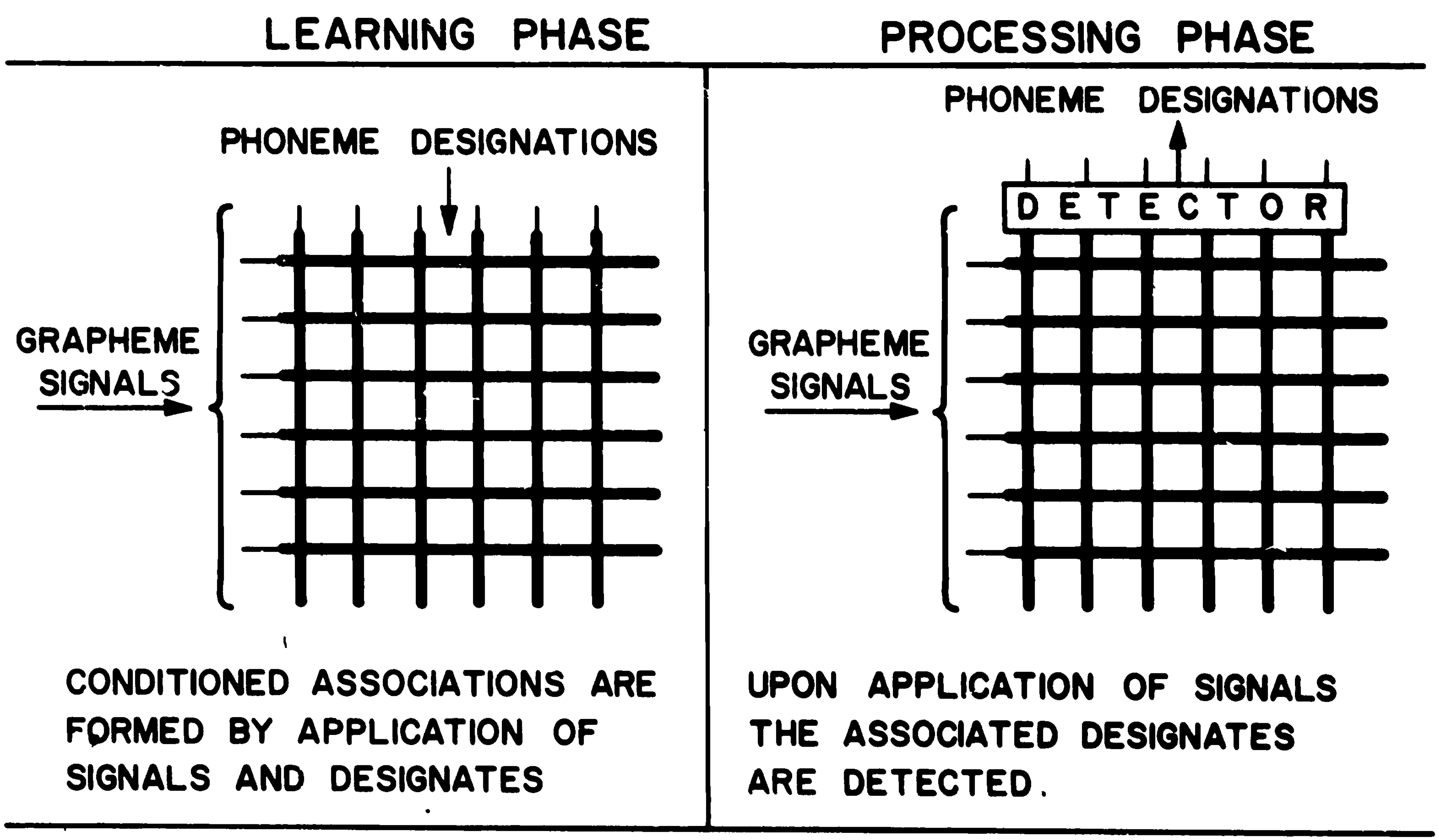


FIGURE 2. PRINCIPLE OF THE LEARNING AND PROCESSING PHASES

The two phases can be described in further detail. The ability to discriminate the input signals and the ability to produce the phonemes (or their numeric equivalents) are resident in the learner and in the machine system. These separate abilities are prerequisite to successful reading but do not actually represent reading ability until the proper conjunctions are affected through the matrix.

At the beginning of the learning phase the matrix cells do not contain any information. In the machine circuit the intersects do not contain any conductivity. In human learning the available phoneme responses are not as yet associated to the printed stimuli.

After the correct grapheme-phoneme associations have been established the system is ready to read. It will continue to read without error as long as the determinants for the phoneme choices are consistent with the determinants which were programmed in the learning phase. The detector illustrated in the processing phase is the computer program in this research which should include the legitimate grapheme-phoneme associations, and the conditions or 'rules' for their association.

The Computer Program

The computer program, identified in this research as Simuread, is an algorithm. In general terms an algorithm is a set of rules which provides a sequence of operations for solving a specific type of problem. There are specific criteria for an algorithm. It must include a finite number of rules or steps which terminate in a solution. The number of rules in Simuread is a function of the word corpus. It is dependent on the number of different grapheme-phoneme associations which occur, and the number of conditions affecting each association. There are 146 hypothesized-legitimate associations in the row matrix previously presented. They are regarded as necessary and sufficient for the conversion of all English orthography to its phonemic representation. The number of conditions which affect the on-off nature of these associations is a far greater, yet finite number.

Each step of an algorithm is definite and unambiguously specified. Obviously an algorithm would not progress toward a solution if the steps were indefinite or ambiguous. The steps or procedures in Simuread are the tests of the word environment which lead to a particular phoneme choice for the grapheme under consideration.

An algorithm is a set of procedures which operate on a pre-specified set of inputs. In Simuread the pre-specified inputs are the letters of the alphabet and punctuation marks.[1]

An algorithm must achieve output with a specifiable relationship to the input. In Simuread the output is the numeric representation of the phonemes. The specific purposes of Simuread are: 1) to analyse the word orthography in terms of its grapheme segments, and 2) to determine the appropriate phoneme representation for each of the segments of orthography.

Finally, an algorithm must be effective. The procedures should be basic, and capable of being carried out in a reasonable length of time. The efficiency of an algorithm is a relative criterion, however, and may be evaluated in terms of the efficiency of humans in carrying out the same procedures.

Model for Computer Simulation of Reading

The concepts and procedures recommended for simulated reading are presented in Figure 3.

(Figure 3)

---

1 The numerals 0 through 9 are also included in Phase III of Simuread; however they are directly associated to the phoneme chains /wən/, /tuw/, etc.

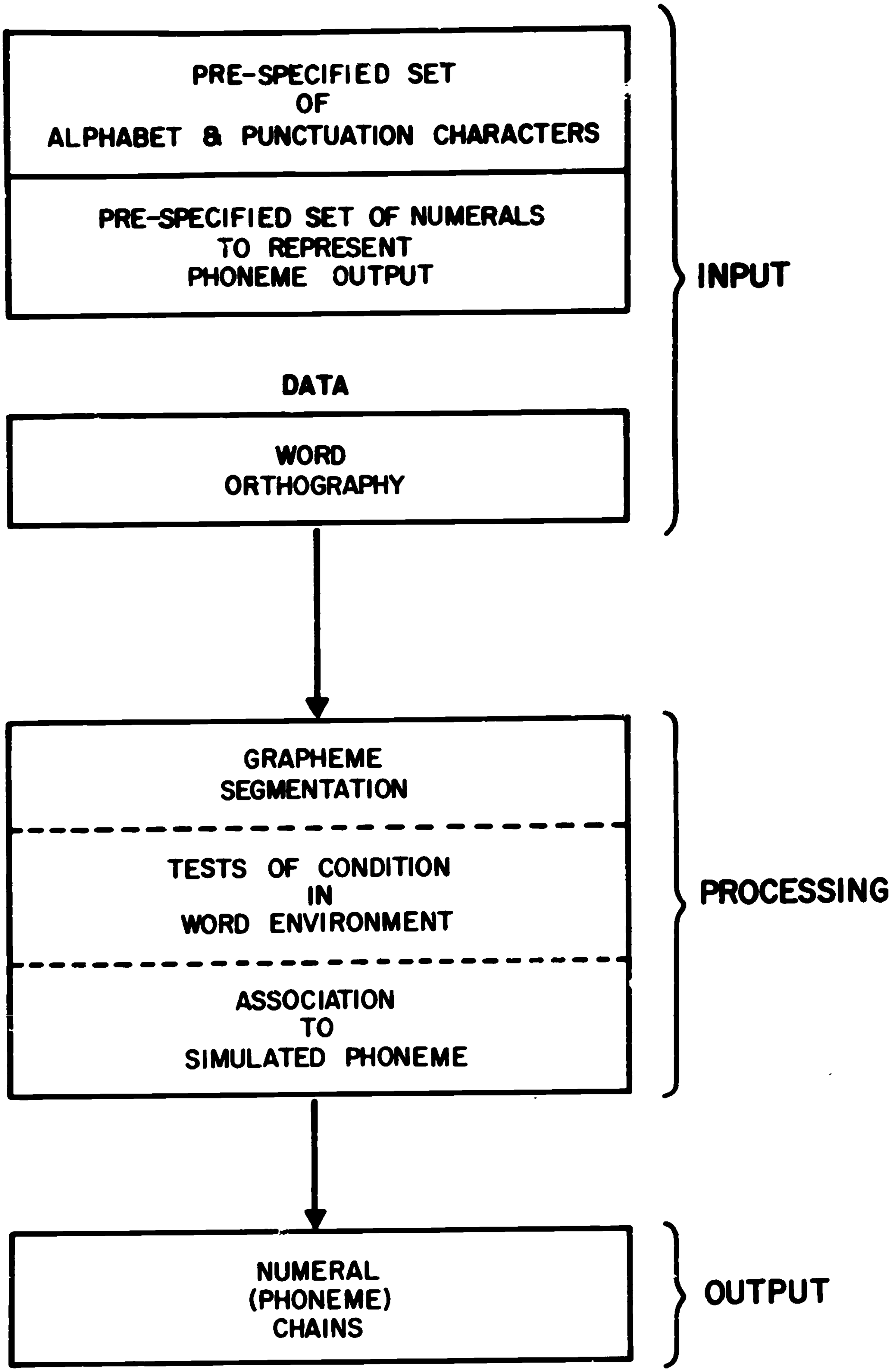


FIGURE 3. COMPUTER SIMULATION OF THE READING PROCESS.

Development of Simuread

One of the problems in initiating this project was for the personnel to achieve a common understanding of the concepts, procedures and purposes. Conferences were held to discuss these, and to establish the necessary communication between the linguistic assistant, the computer programmer and the project director. The progress at the beginning of the project was relatively slow. The assistant in linguistics did a phonemic transcription of all of the words in the pre-primer and primer, and then grouped the words on the basis of structural similarities. The rule statements for these corpora were intentionally simple; presuming a naive learner without prior knowledge of grapheme phoneme associations. They served however to establish some of the procedures to be followed in the program.

A linguistics consultant was retained to assist in the development of rules, and data processing procedures were used to alphabetize the words in each of the succeeding readers, grades 1-6. These procedures facilitated the linguistics effort; however, the rules for the 1-1 and 1-2 readers were extensive, requiring a sixfold increase in the size of the program. So, the 1-1 and 1-2 phases of the program were not complete at the time of this report. The rules characteristics of Simuread and other details of the program are described in this section.

The publisher of the readers presents an outline of the linguistic skills developed at each grade level. The phonetic skills, indicated for the first three books, were extracted from the outline as follows:

Pre-primer: Short vowel sounds
Consonants (m n r s d t p c g h f)
Blends (nd sp nt gr dr mp)
Capitals, period, comma, question mark
Combine sounds to form words, sentences, stories, poems

Primer: Long sounds of vowels
Vowels modified by (r w l ll ff)
Vowel digraphs
Final sounds (ed er le ck nk)
Consonants (b w l k j v)

Book 1-1: Consonants (x y x, soft c and g)
Blends (sh ch tch th wh qu)
Endings (ng ing y ay ey)
Vowel sounds (oo - long and short)
Diphthongs (ow ou oi oy)

The list of phonetic skills appears to be brief; however, it fails to convey the burden of associational learning, particularly in the six readers 1-1 through 3-2. Inspection of the corpora, and of the rules list disclose the level of difficulty. The gradient for the development of the grapheme-phoneme association appears to be steeper than in other basic reading series. The rate of appearance of new words at each grade level is indicated in the following graph. Word redundancy rises sharply after the 3rd grade level.

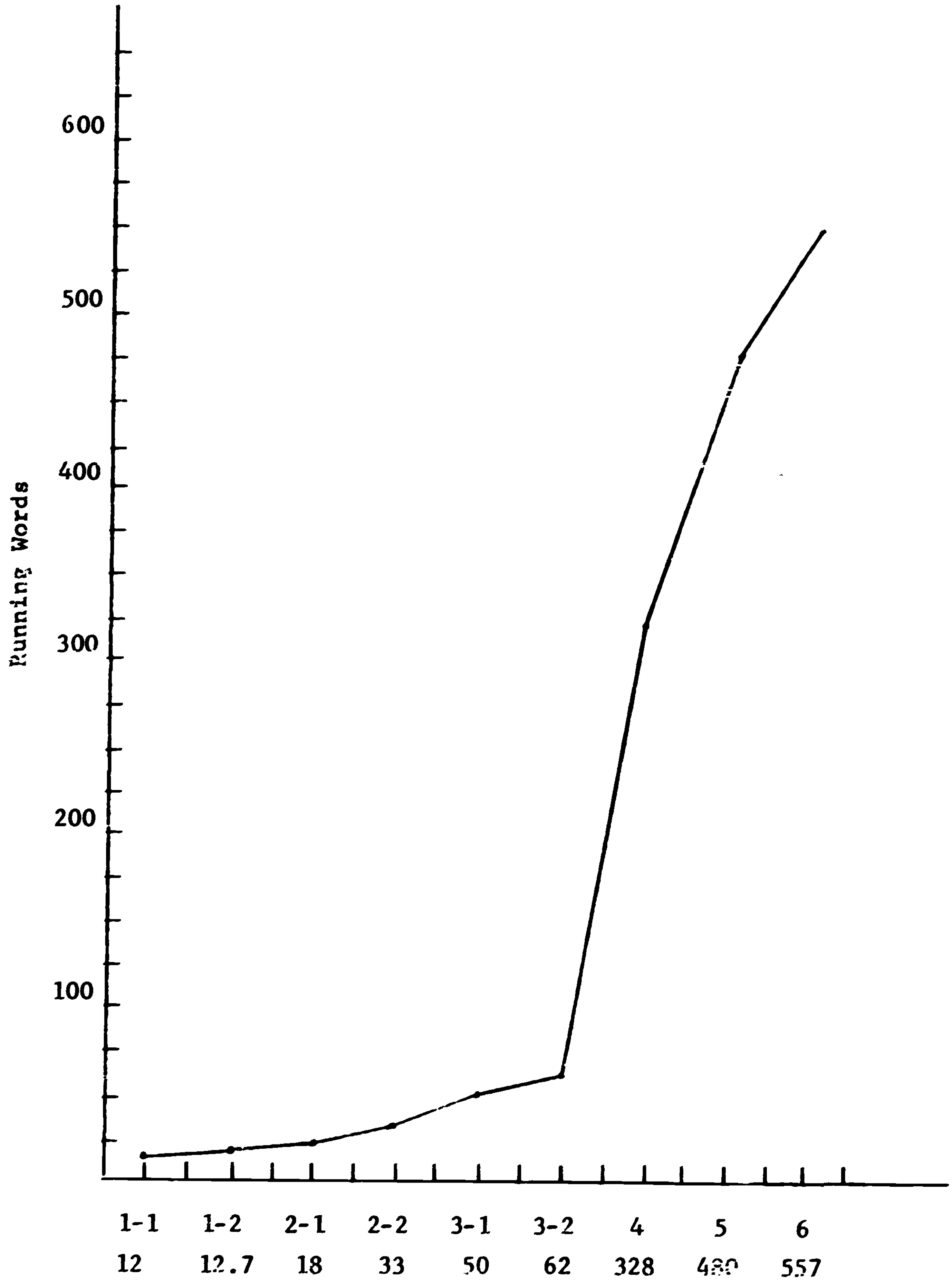


Figure 4. Running words to find one new word in each successive reader.

Reading Skill and Computer Processing

A natural procedure in developing automated reading would be to simulate a child's acquisition of reading skills. It is well known that children can generally learn to read through any of a variety of graded readers. There are differences in basic reading series in the levels at which certain grapheme-phoneme associations are presented, and also in the order of presentation for the various phoneme choices for a given graphic unit. Other variables such as the redundancy of grapheme-phoneme associations, and the frequency of their appearance in different and in repeated words also vary in basic reading series. Comparisons of reading series on these variables have not as yet been made.

A child's response-discrimination abilities are of critical importance in learning to read. These abilities are obvious in the correct reading of words such as said and paid, have and save, gone, done, and hone, some and home, and ache, achieve, chore, chord. The point of view held for this study is that the determinants of phoneme choices are almost always in the word environment. Some of the well-known exceptions to this generalization are the syntax determinants for pronunciations of the present and past tenses of <read> and for the noun and verb forms of <does> and <lead>.

Developing a computer program for reading therefore requires tests for the presence of conditions in the word environment as these are necessary to designate the correct phoneme. A child uses conditional association learning in a similar manner to derive his response discriminations.

Prior to this basic process there is an antecedent requirement that the word orthography be segmented into minimum units. There are two kinds of graphic units which Bloomfield, Barnhart, Gibson, and others have loosely referred to as graphemes. These are: 1) the smallest units which receive independent phoneme associations and 2) the units which determine the phoneme associations of other graphic units, and which in turn may or may not receive phoneme associations. To explain segmentation simply, an example of each type of graphic unit follows. In the words grief and tried the vowel digraph <ie> is identified as a minimum graphic unit. The letters in a vowel digraph do not each receive a phoneme association, but are considered to function as a graphic unit. In the word diet, however, each of the adjacent vowels <i> and <e> are graphemes. The second type of graphic unit is illustrated by the <e> in rate and rated. This segment determines the association for <a> and then receives a non-phoneme or phoneme association.

In human learning the printed word represents a stimulus-complex. A successful analysis of this complex leads to the discrimination of each stimulus. This is similar to the segmentation process in the program.

The input and output components of the computer system are also parallels of the receptive and expressive abilities of the learner. These machine capabilities, and the receptive and oral language abilities of the child exist prior to, and are prerequisites for, their learning to read.

Both machine and human reading are conceptualized as the processing of word data, and the computer program was intended to simulate the progressive development of skill. As stated previously the letters of the alphabet and the numeric representation of phonemes were assumed as the 'givens' of orthography and oral language.

Writing the association rules and the computer program for the primer corpus was a relatively simple task. The consistencies in the vowel associations enabled a high level of accuracy. In processing the 190 primer-word corpus there were only 7 words in which a faulty association occurred. These were: us, runs, Gus with <s> errors and from, front, Don and Tom with <o> errors. The latter errors were because the student assistant had confused the phoneme numerics. The <s> in as, his, has, miss, mess, and cross were processed correctly. Phase I of Simuread achieved a 96% accuracy in processing the 190 words which were used for its construction. Output is appended.

The project then proceeded to the development of Phase II Simuread, i.e., the program for reading the primer. The flow chart and listing of Phase II is included in the appendix. To process the primer corpus correctly required certain sophistications, e.g., the segmenting of diphthongs, mapping the final <e> to the non-phoneme /∅/, etc. These were successfully added to the program; however, the previous difficulties were not removed.

Phase II of Simuread successfully processed 654 of the 770 primer-word corpus. The output is included in the appendix. The rule inadequacies were: 1) mapping the <e> in words such as aimed and raised to the non-phoneme, 2) mapping the <s> to /z/ after voiced consonants, and 3) incorrect choices for certain medial vowels and initial vowels. The <l> <e> in words such as apple and rattle were correctly mapped to /l/ and /∅/. This is consistent with Kenyon and Knott's pronunciation, whereas in phonemic analysis of oral language this would probably be transcribed to /ɔl/.

In judging the performance of Phase II for the correct processing of 85% of the word corpus there may be conflicting opinions. Obviously a few changes could be easily effected to produce about 95% accuracy. There are only a few words such as have, vacant, and corral in this corpus which would be exceptions to the more general rules. The attainment of 85% accuracy for the primer corpus is not presented as an excellent performance, nor a poor performance. Some of the errors which occurred represent faulty generalization. For example, the procedure which led to accurate processing of are as /06 32 50/ was misapplied to mare, flare, dare, etc. This misapplication of prior learning can also occur when children learn to read. In contrast to this, the final <s> to /z/ after voiced consonants is characteristic of the oral language and this error rarely occurs in learning to read.

The development of Phase III to process the 1-1 and 1-2 corpora became a lengthier technical problem. Developing a more sophisticated program does not simply involve adding more rules; it also requires the revision, or refinement, of prior rules. The hierarchical arrangement of rules also becomes important not only for accurate processing, but also for efficient programming. The generality, or the degree of applicability of a given rule also enters into the strategy of the program. The strategy for computer processing might attempt to imitate the generalizations which a successful first grade reader is presumed to possess after reading the two texts. These may not be apparent, however, until the rules for the associations are explicitly known and classified.

Writing the program for the 1-1 and 1-2 readers is underway at the time of this report. It will probably not be operational for another six months.

Testing the Program

The adequacy of the program was tested through processing the three- four- and five-letter words in the Rinsland vocabulary list (1945). There were 320 three-letter words, 1049 four-letter words and 1332 five-letter words in the vocabulary list.

The processing of the three-letter words achieved 71% accuracy. There were 95 word-errors in the 320 word corpus. Most of these errors occurred because there was a lack of rules to make the necessary conversions. In a few cases the necessary rule existed in the program but was not applied because the previous rules in the hierarchy processed the error choice. Following is a list of the conversions which were not made in the error words:

Vowels

final <e> → 50*

*Note: This rule existed in the program, and was functioning for /aCe#, but was erroneously mapped to /02/ for other preceding vowels.

final <o> → 40, 39
final <i> → 35
<a> → 09, 02, 06
<e> → 02, 35
<o> → 06 /_g
→ 09 /_r
→ 05 /_n
<u> → 39, 42, 47
<y> → 35, 37

Consonants

<c> → 23
<g> → 17
<h> → 50
<s> → 25

Consonant digraphs

<sh> → 24
<th> → 22
<wh> → 20

Vowel-semi vowel digraphs

<ay> → 36
<aa> → 06
<ea> → 01 /_r
<ew> → 39, 42, 40
<oa> → 09 /←r
<oi> → 38
<oo> → 39

Some of the needed conversions, e.g. <i> → 35 for <ski> and <aa> → 06 for <baa>, pertained to only one word in the corpus; whereas the final <e> → 50 pertained to 11 of the 95 word errors.

There were 577 errors in processing the 1029 four-letter word corpus. The program only achieved a 45% accuracy in the word output. A few of the word errors in this corpus were due to faulty segmentation of adjacent vowel graphemes, e.g. area, duet, liar, lion, poem, poet, quit, quiz, ruin, suet, view. The other errors were due to the inadequacy of the rules for the phoneme selections.

<u>Vowels</u>

<a> → 36
final <e> → 36 <cafe>
<e> → 35, 01, 36
<i> → 37, 05
<o> → 40, 07
<u> → 01, 05, 31, 33
<y> → 01

<u>Consonants</u>

<b>, <d>, <g>, <z> → 50
<l> → 50 /_f
l_k
<n> → 50 /m_
<t> → 50 /_ch
<w> → 50 /_r

<u>Consonant digraphs</u>

<ch> → 12, 16
<gh> → 50
<wh> → 31

<u>Vowel-semi vowel digraphs</u>

<ay> → 02 <says>
<au> → 03, 09
<ea> → 35, 02
<eau> → 40
<ee> → 01
<ey> → 37
<oe> → 39 <shoes>
<oo> → 07, 09
<ou> → 09, 40, 07
<ie> → 01 <pier>
<ue>, <ui> → 39

There were 951 word errors in the 1332 five-letter word corpus. The proportion of complete words processed correctly by the program was only 29 percent. The final <e> → 50 and final <s> → 25 rules were operational in processing some of the five-letter words. This however did not improve the accuracy of word output, because of the prevailing errors in other phoneme choices. In addition to the inadequacies in the three- and four-letter word corpora the following conversions were not correctly processed:

Vowels

<a> → 05
<e> → 36 <crepe>
<o> → 39 <prove>
<y> → 05 <myrrh>

Consonants

<m> → 50
<n> → 29 <angel>

Consonant digraphs

<ce> → 24 <ocean>
<ph> → 18
<gh> → 15 <ghost>

Vowel-semi vowels digraphs

<ie> → 35
<oo> → 05
<ui> → 01 <build>

Although the word-error in this output is high the number of additional rules unique to this corpus is relatively low.

Statistical Analysis of an Association Matrix

Analyses of the components of English have usually centered on two separate systems, words and sounds. Perhaps the most outstanding of such research is Roberts' (1965) frequency analysis of the segmental phonemes occurring in the 10,065 words in Horn's *Basic Writing Vocabulary*. The phonemic transcription of this vocabulary yielded 66,534 segmental phonemes. Examples of word frequency studies are Thorndike's, Horn's and Rinsland's vocabulary lists.

It was intended in this research to analyze the matrix of associations occurring at each grade level in the Lippincott series. The counting of the associations and computation of percentages of joint occurrence was to be incorporated into the processing program. This was not included at the time of this report, however, so the association matrix from another corpus is used to illustrate a procedure for analyzing an association matrix.

A study in which the grapheme-phoneme associations were reported (Leton, 1966) presents the matrix for 6949 running words in the primer and pre-primer of the Ginn basic reading series. A copy of the data matrix is available from the author. There was a total of 23,247 associations in the 6949 words. An index of grapheme similarity was determined from the proportion of shared phonemes. For example, in this corpus the grapheme <c> appeared 333 times and it was associated to the phoneme /k/ in each instance. There were 376 occurrences of <k> of which .923 were associated with /k/ and .074 to Ø. The graphemes <k> and <c>, therefore, have a common pronunciation for 92% of the <k> occurrences. Of the 65 possible graphemes 43 occurred in the primer and pre-primer corpus and 31 of these shared phoneme representation.

The matrix of similarity among graphemes using the function of overlap was computed as follows:

$$A_{ij} = \sum_{k=1}^{43} [P_{ik} \text{ or } P_{jk}, \text{ whichever is smaller}]$$

where P_{ik} is the proportion in which grapheme i is associated with phoneme k.

The similarity matrix is presented in table A. The matrix was analyzed using Ward's (1963) hierarchical grouping procedure. The grapheme groups obtained after 17 iterations are presented in figure 4. The sets of graphemes were formed in hierarchical order by assembling those which resulted in the smallest contribution to the total within-group sums of squares. The total variation among the grapheme pairs (SSE) was minimally increased at each grouping step. The area of each of the bars in the graph indicates the ratio of the within-group to the total group deviation. The eight groups included 25 graphemes and accumulated 24.8% of the total deviation.

The common phonemes shared in each of the groups is as follows: <c> <ck> <k> <q> → /k/; <g> <l> <t> <b> → ∅ (note: the second of a double consonant was associated to the non-phoneme ∅)· <s> <z> → /z/; <y> <ee> <ea> → /iy/; <a> <o> → /a/, <e> <u> <i> → /ə/ and /u/, <ey> <ai> <ay> → /ey/; and <ou> <oo> <ue> <ew> → /uw/. The single members, e.g. <m> <ng> <r> <th>, did not share phonemes with other graphemes.

This analysis illustrates a procedure for identifying the degree of similarity among graphemes on the basis of their phoneme associations. The alternative analysis, i.e., identifying the similarity of phonemes on the basis of their grapheme representation could also be carried out. For example, the phonemes /e/, /ə/, /u/, /iy/ and /∅/ would be grouped because of their common representation by the <e> grapheme. The analysis would

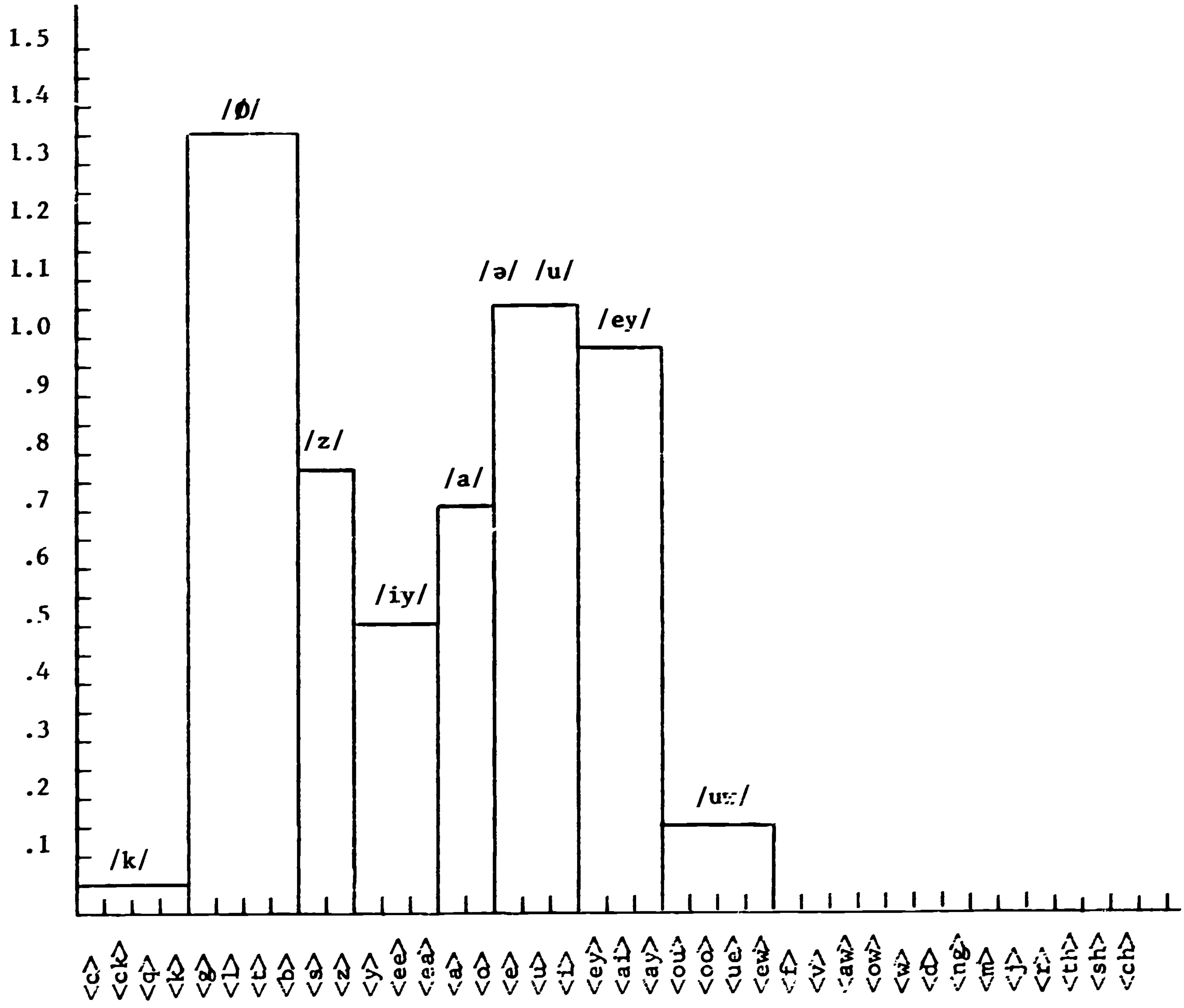


Figure 4. A presentation of grapheme groups formed on the basis of their common association to phonemes.

provide a quantitative estimate of the strength of such associations. Factor analyses of the association matrices produce results similar to the grouping procedure.

In a previous study (Dunn-Rankin and Leton 1968) hierarchical grouping was used for comparing two indices of the similarity of lower case letters. Factor analytic procedures were used to derive groups of letters with structural congruency (Dunn-Rankin, et al. 1968). Although these were analyses of a specific graphic representation (Century Schoolbook font) the statistical procedures are also applicable to grapheme-phoneme association matrices.

Additional Progress

Rules have been written for the other readers, i.e., 2-1, 2-2, 3-1, 3-2, 4, 5, and 6. These corpora have also been keypunched and alphabetic lists prepared to facilitate the identification of new words. These represent further work on this project not included in this report. We were unable to begin writing the program for these rules, and their inclusion in this report would make it unduly long.

Further insights were gained from these efforts. For example, names, names of geographic places, contractions, and words of foreign origin present difficulties. Following is a list of words from the 2-2 reader which illustrate this problem. The graphemes which present difficulty, i.e. require associations different from those which would be obtained from the body of rules, are underlined.

Amelia	English	I'd	Pasteur
Betsy	Europe	I'll	pasteurization
Boston	ocean	Louis	Russia
Crimea	Friday	Louisa	
David	Garcia	Manuel	Susan
Davis	George	Massachusetts	Sweden
Derbyshire	Hawaii	potatoes	women
don't			won't

The compound words present a definite problem in readers 4, 5, and 6. The grapheme segmentation for words such as: uphill, baseball, and rosebush would be incorrect, and phoneme choices would be made for the <ph> and <e> graphemes. To process these correctly would require word memory. This problem could be eliminated however by placing a restriction on the orthographer to insert word boundary in all compound words.

Another difficulty which arises as the syllabic length of words increase is the difficulty of ignoring stress placements which affect vowel reduction and produce elision.

In readers 4-6 about six percent of the sample of new words would be processed incorrectly on the basis of the rules for grapheme-phoneme associations produced to that level. The significance of this percent is not interpretable because the comprehensibility of output cannot be tested.

Conclusion

The primary purpose of this research was to develop a computer program to convert English word orthography to appropriate phoneme sequences. An information processing model was theorized in which the conditional association of phonemes to graphic units was regarded as the critical process in reading.

The feasibility of automated reading may be discussed. Linguists have sometimes expressed the opinion that reading of traditional English orthography could not be automated because of the complexity of the rules and the number of conditions which determine the phonemic representation. In contrast to this, educators can point out that primary grade pupils are generally successful in learning to read, i.e., in learning to convert the word orthography to phonic English. And teachers are generally successful in teaching these skills with a variety of methods and materials.

The feasibility of automated reading should be discussed as a relative question. There are a finite number of grapheme-phoneme associations in any corpus. The conditions, or rules which govern the associations in any particular text, however, are not generally available. A certain number of rules will result in a certain degree of success in automated reading. There are technical problems involved in segmenting the orthography, in the efficient ordering of rules, and in determining the environmental boundaries, e.g., word, syntax and semantic boundaries, which govern the graphic-phonic associations. The level of success, or the degree of accuracy in phoneme output is related to these technical problems.

The segmental phoneme output obtained in this program may have little practical value. A segmental phoneme chain without stress features and other suprasegmental phonemes is not comprehensible as oral language. On the other hand, the stress phonemes could be superimposed on the segmental phonemes, using a 'piggy-back' program. Additional rules for the levels of word stress in sentences and for clause terminals also could be programmed. The phonemic output could then be used to generate synthetic speech, and the comprehensibility of the synthetic speech could be tested. These further developments are necessary before any practical value may be realized.

The errors which appear in automated reading are both similar to, and different from, those made by a beginning reader. For example, in the primer corpus the <s> in us was associated to /z/ because of the rule for is and has. The words east and feast were processed correctly but the <s> in case was processed as /s/. These errors represent a misapplication of the rules in automated reading; whereas, a beginning reader who is a 'native' speaker of English would not make such errors. Other errors, particularly in the choices of vowel and diphthong phonemes, are similar for the beginning reader and the program. The computer program does not produce reversal errors, e.g., was for saw, dig for big, etc. These are faulty sequencing of output, and faulty perception of input, which may appear for beginning readers but do not appear in automated reading.

The significance of the types of errors can also be discussed. Obviously, certain errors tend to degrade the quality of the output very markedly. If a grapheme should be associated to the nonphoneme, but through error is associated to a segmental phoneme the output sequence is faulty. For example, in the primer programs the <e> in called was erroneously mapped to /e/. The rule that correctly processed <e> in braided produced the error in called. The error of /s/ for /z/ in raise is also interpreted as a significant error, whereas confusions of /s/ and /z/ in other words would not be as important. Several criteria are appropriate to judge the degree of accuracy. These would include: 1) the number of incorrect grapheme-phoneme associations, 2) the ratio of the misapplication of rules to the correct application, and 3) the extent to which the quality of the phoneme output is degraded by particular errors.

The computer simulation of reading enables a more objective classification of reading errors. The errors in grapheme-phoneme associations can be differentiated from: 1) faulty perception of the graphic features, 2) faulty segmentation of the orthography, 3) faulty sequencing in either the processing or the speech production of the phoneme chain.

There appear to be two levels of rules development in writing the computer program. The first level, for consonants and consonant digraphs, is relatively easy. There are fewer and more consistent associations for these units. The second level of rules development refers to the vowels and the vowel-semivowel digraphs. The task of programming rules for each of the vowel and vowel-semivowel associations is as extensive as for all the consonant associatious combined. The beginning reader, and the program for the pre-primer, does not recognize that the vowel and consonant graphic units have different potentials for phoneme representation. They also differ in the complexity of the phoneme decisions. This recognition accumulates with the association experiences. There are implications for the strategy of programming, and also for the amount of effort for the two levels of rule development.

In evaluating this study the amount of progress made in simulating the associational learning necessary for reading may be judged as disappointing. The progress is estimated as representing about 20 to 25 percent of the mastery of grapheme-phoneme associations necessary for primary reading skills. There were practical problems that prevented better progress. These included scheduling the personnel effort, using efficient data processing to facilitate the preparation of rules, and problems in interdisciplinary communication.

One of the potential contributions to the methodology of reading research is the statistical analysis of the grapheme-phoneme association matrix. Hierarchical grouping procedures were used for this analysis; however, factor analyses of association matrices would also produce similar results. Studies of phoneme frequency, and word counts of oral and written vocabulary are illustrative of linguistics and language research. These are typically in the oral or printed representation of the language. A procedure for analyzing the relationship between a particular text or printed word corpus and its phonemic representation should result in a more objective analysis of the reading process. Judgments on the similarities and differences among beginning reading books in various series can be made by comparing the grapheme-phoneme association matrices. The redundancy and infrequency of certain associations would be apparent, and such data should have implications for the efficient development of reading skills.

References

Bloomfield, L. and Barnhart, C. Let's Read: A Linguistic Approach. Detroit: Wayne State Univ. Press. 1961.

Dunn-Rankin, P. and Leton, D. A. Hierarchical grouping procedures for comparing indices of letter similarity. Perceptual & Motor Skills. 1968, 27, 457-458.

Dunn-Rankin, P., Leton, D. A. and Shelton, V. F. Congruency factors related to visual confusion of English letters. Perceptual & Motor Skills. 1968, 26, 659-666.

Gibson, E. J., Rich, A., Osser, H. and Hammond, M. The role of grapheme-phoneme correspondence in the perception of words. Amer. J. of Psychology. 1962, 75, 4, 554-570.

Gleason, H. A. English Stress and Intonation: An Introduction to Descriptive Linguistics. New York: Holt, Rinehart and Winston. 1961.

Hanna, P. R., Hanna, J. S., Hodges, R. E. and Rudorf, E. H. Phoneme-grapheme correspondence as cues to spelling improvement. U.S.O.E. Research Bureau. 1966, 133 pp.

Horn, E. A. A Basic Writing Vocabulary: 10,000 Words Most Commonly Used in Writing. University of Iowa Monographs in Education, First Series, No. 4, Iowa City, 1926.

Judd, C. H. Reduction of articulation. Amer. J. Psychology. 1927, 39, 313-322.

Leton, D. A. Oral reading as grapheme segmentation and phoneme choice. Amer. Psychologist. 1967, 22, 545-546.

Leton, D. A. Conditional probability in the phonemic representation of English words. In J. Figurel (Ed.), Forging Ahead in Reading. Newark, Delaware: International Reading Association. 1968, 12, 1, 567-570.

Rinsland, H. D. A Basic Vocabulary of Elementary School Children. New York: Macmillan Co. 1945, 636 pp.

Roberts, A. H. A Statistical Linguistic Analysis of American English Words. London: Moriton & Co. 1965, 437 pp.

Rudorf, E. H. Measurement of spelling ability. Linguistic Cues for Program of Spelling Curriculum. Newark: University of Delaware Press. 1964.

See, R. Mechanical translation and related language research. Science. 1964, 144, 3619, 621-626.

Steinbuch, K. Die lernmatrix. Kybernetik. 1961, 1, 36-45.

Stratemeyer, C. G. and Smith, H. L. The Linguistic Science Readers. Evanston: Harper & Row. 1963.

Trager, G. L., and Smith, H. L. An outline of English Structure. Oklahoma: Battenburg Press. 1961.

Vanezky, R. L. A Study of English Spelling to Sound Correlations. Berkeley: University of California. 1962.

Ward, J. H. and Hook, M. S. Application of an hierarchical grouping procedure to a problem of grouping profiles. Journal of Ed. & Psych. Meas. 1963, 23, 69-81.

Weir, R. H. Formulation of grapheme-phoneme correspondence rules to aid in the teaching of reading. Cooperative Research Report No. S039. Stanford University. 1964.

Appendix

Table A. Matrix of Percent Overlap in Phoneme Designates

Computer Program: Simuread Phase II

Output Pre-Primer Corpus

Output Primer Corpus

Table B. Pre-Primer and Primer Rules

Flowchart for Program

Publisher Outline of Phonetic Skills for Lippincott series

Table A

Matrix of Percent Overlap in Phoneme Designates

Graphemes	c	ck	d	f	g	h	k	l	n	ng	p	q	s	t	v	w	y
c	1.000																
ck	1.000	1.000															
d	0.0	0.0	1.000														
f	0.0	0.0	0.0	1.000													
g	0.0	0.0	0.002	0.0	1.000												
h	0.0	0.0	0.002	0.0	0.033	1.000											
k	0.923	0.923	0.002	0.0	0.074	0.033	1.000										
l	0.0	0.0	0.002	0.0	0.111	0.033	0.074	1.00									
n	0.0	0.0	0.002	0.0	0.029	0.029	0.029	0.029	1.000								
ng	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	1.000							
p	0.0	0.0	0.002	0.0	0.051	0.033	0.051	0.051	0.029	0.0	1.000						
q	1.000	1.000	0.0	0.0	0.0	0.0	0.923	0.0	0.0	0.0	0.0	1.000					
s	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000				
t	0.0	0.0	0.007	0.0	0.111	0.033	0.074	0.130	0.029	0.0	0.051	0.0	0.0	1.000			
v	0.0	0.0	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000		
w	0.0	0.0	0.002	0.0	0.003	0.003	0.003	0.003	0.003	0.0	0.003	0.0	0.0	0.003	0.0	1.000	
y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000
z	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.213	0.0	0.0	0.0	0.0
a	0.0	0.0	0.002	0.0	0.018	0.018	0.018	0.018	0.018	0.0	0.018	0.0	0.0	0.018	0.0	0.003	0.0
e	0.0	0.0	0.002	0.0	0.111	0.033	0.074	0.151	0.029	0.0	0.051	0.0	0.0	0.130	0.0	0.003	0.116
i	0.0	0.0	0.002	0.0	0.111	0.033	0.074	0.151	0.029	0.0	0.051	0.0	0.0	0.130	0.0	0.003	0.167
o	0.0	0.0	0.002	0.0	0.009	0.009	0.009	0.009	0.009	0.0	0.009	0.0	0.0	0.009	0.0	0.003	0.0
u	0.0	0.0	0.002	0.0	0.111	0.033	0.074	0.151	0.029	0.0	0.051	0.0	0.0	0.130	0.0	0.181	0.0
ee	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.422
ea	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.116	0.047	0.0	0.0	0.0	0.0	0.422
ey	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.287
ai	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ay	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.118
aw	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ou	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
oo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Graphemes	c	ck	d	f	g	h	k	l	n	ng	p	q	s	t	v	w	y
ue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b	0.0	0.0	0.002	0.0	0.101	0.033	0.074	0.101	0.029	0.0	0.051	0.0	0.0	0.101	0.0	0.003	0.0
ew	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Graphemes	z	a	e	i	o	u	ee	ea	ey	ai	ay	aw	ou	ow	oo	ue	b	ew
z	1.000																	
a	0.0	1.000																
e	0.0	0.203	1.000															
i	0.0	0.018	0.220	1.000														
o	0.0	0.296	0.023	0.009	1.000													
u	0.0	0.106	0.229	0.173	0.010	1.000												
ee	0.0	0.005	0.116	0.047	0.0	0.0	1.000											
ea	0.0	0.022	0.116	0.041	0.0	0.0	0.995	1.000										
ey	0.0	0.022	0.116	0.221	0.0	0.0	0.174	0.169	1.000									
ai	0.0	0.022	0.0	0.0	0.0	0.0	0.005	0.0	0.502	1.000								
ay	0.0	0.0	0.0	0.174	0.0	0.0	0.0	0.0	0.328	0.0	1.000							
aw	0.0	0.005	0.0	0.0	0.013	0.0	0.0	0.0	0.0	0.0	0.0	1.000						
ou	0.0	0.0	0.0	0.0	0.265	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000					
ow	0.0	0.0	0.0	0.0	0.130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.171	1.000				
oo	0.0	0.0	0.0	0.0	0.298	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.829	0.033	1.000			
ue	0.0	0.0	0.0	0.0	0.265	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.829	0.0	0.967	1.000		
b	0.0	0.018	0.101	0.101	0.009	0.101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000	
ew	0.0	0.0	0.0	0.0	0.265	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.829	0.0	0.967	1.000	0.0	1.000

```
C       SIMULATED READING PROGRAM
        DIMENSION GRAPH(28), LPHONE(28), DATA(80), LSW(80)
        READ(5,1) (GRAPH(I), I=1,28)
      1 FORMAT (28A1)
        WRITE(6,200) (GRAPH(I), I=1,28)
    200 FORMAT(1X, 6HALPHA , 28A1)
        READ(5,2) (LPHONE(I), I=1,28)
      2 FORMAT (28I2)
        WRITE(6,201) (LPHONE(I), I=1,28)
    201 FORMAT(1X, 8HNUMERIC , 28I2)
        READ(5,7) MM
      7 FORMAT (I3)
        DO 2000 LG=1,MM
        LCOUNT = 1
        READ(5,3) (DATA(I), I=1,80)
      3 FORMAT (80A1)
        J = 2
        JA = 1
        JAA = 1
        LJ = 0
    100 CONTINUE
        DO 10 I=J,80
        K = I+1
        LF = JA
        LLC = LCOUNT+LF-2+JAA
        IF (DATA(I).EQ.GRAPH(27)) GO TO 20
        LCOUNT = LCOUNT + 1
     10 CONTINUE
        GO TO 2000
     20 CONTINUE
        LLD =LLC
        LC = LF
        LD = LF
        GO TO 40
C       TH AND SS LETTER ADVANCE
   1001 LF = LB+1
        LC = LC+1
        GO TO 40
C       VOWELS LETTER ADVANCE
   1002 LF = LB
        LC = LC+1
     40 DO 12 I=LF,LLC
        DO 11 M = 1,26
        LM = M
        LA = I-1
        LB = I+1
        LH = LLC-I
C       TH TEST
```

```
           IF (DATA(I).NE.GRAPH(20)) GO TO 1010
           IF (DATA(LB).NE.GRAPH(8)) GO TO 390
           LLD = LLC-1
           LSW(LC) = 19
           GO TO 1001
       390 IF (DATA(LB).NE.GRAPH(20)) GO TO 1000
           LSW(LC) = 11
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
     C     S TEST
      1010 IF (DATA(I).NE.GRAPH(19)) GO TO 1020
           IF (DATA(LB).NE.GRAPH(19)) GO TO 1011
           LSW(LC) = 23
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
      1011 IF (DATA(LA).EQ.GRAPH(1)) GO TO 1012
           IF (DATA(LA).EQ.GRAPH(5)) GO TO 1012
           IF (DATA(LA).EQ.GRAPH(9)) GO TO 1012
           IF (DATA(LA).EQ.GRAPH(15)) GO TO 1012
           IF (DATA(LA).EQ.GRAPH(21)) GO TO 1012
           GO TO 1000
      1012 IF (I.NE.LLC) GO TO 1000
           LSW(LC) = 25
           GO TO 12
     C     A TEST
      1020 IF (DATA(I).NE.GRAPH(1)) GO TO 1030
           IF (LF.EQ.LLC) GO TO 312
           GO TO 313
       312 LSW(LC) = 05
           GO TO 1002
       313 IF (DATA(LB).EQ.GRAPH(9)) GO TO 314
           IF (DATA(LB).EQ.GRAPH(23)) GO TO 315
           IF (DATA(LB).EQ.GRAPH(19)) GO TO 316
           IF (DATA(LB).EQ.GRAPH(4)) GO TO 317
           IF (DATA(LB).EQ.GRAPH(16)) GO TO 318
           IF (DATA(LB).EQ.GRAPH(20)) GO TO 319
       730 IF (DATA(LB).EQ.GRAPH(14)) GO TO 723
       733 IF (DATA(LB).EQ.GRAPH(18)) GO TO 724
           IF (LH.EQ.2) GO TO 725
       735 IF (LH.EQ.3) GO TO 726
       736 IF (DATA(LA).EQ.GRAPH(27)) GO TO 727
           L = 1
           GO TO 300
       310 IF (DATA(LB).EQ.GRAPH(27)) GO TO 1000
       311 LSW(LC) = 03
           GO TO 1002
```

```
      314 LI = LB+1
          IF (DATA(LI).NE.GRAPH(18)) GO TO 728
          LLD = LLC-1
          LSW(LC) = 02
          GO TO 1001
      728 LLD = LLC-1
          LSW(LC) = 36
          GO TO 1001
      315 LLD = LLC-1
          LSW(LC) = 09
          GO TO 1001
      316 LI = LB+1
          IF (DATA(LI).NE.GRAPH(19)) GO TO 729
          LSW(LC) = 03
          GO TO 1002
      729 IF (DATA(LA).NE.GRAPH(23)) GO TO 730
          LSW(LC) = 06
          GO TO 1002
      317 LI = LB+1
          IF (DATA(LI).NE.GRAPH(4)) GO TO 730
          LSW(LC) = 03
          GO TO 1002
      318 LI = LB+1
          IF (DATA(LI).NE.GRAPH(16)) GO TO 730
          LSW(LC) = 03
          GO TO 1002
      319 LI = LB+1
          IF (DATA(LI).NE.GRAPH(20)) GO TO 731
          LSW(LC) = 03
          GO TO 1002
      731 IF (DATA(LA).NE.GRAPH(23)) GO TO 730
      732 LSW(LC) = 06
          GO TO 1002
      723 IF (DATA(LA).EQ.GRAPH(23)) GO TO 732
          GO TO 733
      724 IF (DATA(LA).EQ.GRAPH(23)) GO TO 1000
          L = 7
          GO TO 400
      734 IF (LH.NE.2) GO TO 735
          LSW(LC) = 02
          GO TO 1002
      725 LI = LB+1
          IF (DATA(LI).NE.GRAPH(5)) GO TO 735
          LSW(LC) = 36
          LJ = 1
          GO TO 1002
      726 IF (DATA(LB).NE.GRAPH(13)) GO TO 736
          LI = LB+1
```

```
          IF (DATA(LI).NE.GRAPH(5)) GO TO 736
          LI = LB+2
          IF (DATA(LI).NE.GRAPH(4)) GO TO 736
          LSW(LC) = 36
          GO TO 1002
      727 LSW(LC) = 03
          GO TC 1002
C         I TEST
     1030 IF (DATA(I).NE.GRAPH(9)) GO TO 1040
          IF (DATA(LB).EQ.GRAPH(5)) GO TO 322
          L = 2
          GO TO 300
      320 IF (LH.EQ.3) GO TO 324
          IF (LH.EQ.2) GO TO 325
      326 IF (DATA(LB).NE.GRAPH(18)) GO TO 323
          L = 5
          GO TO 400
      321 LSW(LC) = 05
          GO TC 1002
      322 LLD = LLC-1
          LSW(LC) = 37
          GO TO 1001
      324 LI = LB+1
          IF (DATA(LI).NE.GRAPH(5)) GO TO 326
          LI = LB+2
          IF (DATA(LI).NE.GRAPH(18)) GO TO 326
      327 LSW(LC) = 37
          GO TO 1002
      325 LI = LB+1
          IF (DATA(LI).EQ.GRAPH(5)) GO TO 327
          IF (DATA(LI).NE.GRAPH(4)) GO TO 323
          LI = LB
          IF (DATA(LI).NE.GRAPH(14)) GO TO 326
          GO TO 327
      323 GO TO 1000
C         E TEST
     1040 IF (DATA(I).NE.GRAPH(5)) GO TO 1050
          IF (DATA(LB).EQ.GRAPH(1)) GO TO 331
          IF (DATA(LB).EQ.GRAPH(5)) GO TO 332
          IF (DATA(LB).EQ.GRAPH(27)) GO TO 334
      701 LI = LB+2
          IF (DATA(LI).EQ.GRAPH(27)) GO TO 335
      703 IF (DATA(LB).EQ.GRAPH(4)) GO TO 336
      705 IF (DATA(LA).EQ.GRAPH(27)) GO TO 337
          IF (DATA(LB).EQ.GRAPH(18)) GO TO 338
      707 L = 6
          GO TO 300
      330 IF (DATA(LB).EQ.GRAPH(27)) GO TO 1000
```

```
0185               LSW(LC) = 02
0186               GO TO 1002
0187          331  LI = LB+1
0188               IF (DATA(LI).EQ.GRAPH(18)) GO TO 339
0189               LLD =LLC-1
0190               LSW(LC) = 35
0191               GO TO 1001
0192          339  LLD = LLC-1
0193               LSW(LC) = 01
0194               GO TO 1001
0195          332  LLD = LLC-1
0196               LSW(LC) = 35
0197               GO TO 1001
0198          334  LE = LC-1
0199               IF (LSW(LE).NE.19) GO TO 700
0200               LSW(LC) = 05
0201               GO TO 1002
0202          700  LI = LA-1
0203               IF (DATA(LI).NE.GRAPH(27)) GO TO 701
0204               IF (DATA(LA).EQ.GRAPH(2)) GO TO 702
0205               IF (DATA(LA).EQ.GRAPH(23)) GO TO 702
0206               IF (DATA(LA).EQ.GRAPH(13)) GO TO 702
0207               IF (DATA(LA).NE.GRAPH(6)) GO TO 701
0208          702  LSW(LC) = 35
0209               GO TO 1002
0210          335  LI = LB+1
0211               IF (DATA(LI).NE.GRAPH(5)) GO TO 703
0212               L = 3
0213               GO TO 300
0214          704  LSW(LC) = 35
0215               GO TO 1002
0216          336  LI = LB+1
0217               IF (DATA(LI).NE.GRAPH(27)) GO TO 705
0218               IF (DATA(LA).EQ.GRAPH(4)) GO TO 706
0219               IF (DATA(LA).NE.GRAPH(20)) GO TO 705
0220          706  LSW(LC) = 01
0221               GO TO 1002
0222          337  LSW(LC) = 02
0223               GO TO 1002
0224          338  L = 13
0225               GO TO 400
0226          708  LSW(LC) = 05
0227               GO TO 1002
0228          333  GO TO 1000
            C      U TEST
0229         1050  IF (DATA(I).NE.GRAPH(21)) GO TO 1060
0230               L = 4
0231               GO TO 300
```

```
      340 IF (DATA(LB).EQ.GRAPH(27)) GO TO 342
          LI = LB+1
          IF (DATA(LB).NE.GRAPH(12)) GO TO 343
          IF (DATA(LI).EQ.GRAPH(12)) GO TO 342
      343 LSW(LC) = 05
          GO TO 1002
      342 LSW(LC) = 07
          GO TO 1002
      341 GO TO 1000
C         G TEST
     1060 IF (DATA(1).NE.GRAPH(15)) GO TO 3000
          IF (DATA(LB).EQ.GRAPH(1)) GO TO 351
          IF (DATA(LB).EQ.GRAPH(5)) GO TO 351
          IF (DATA(LB).EQ.GRAPH(23)) GO TO 352
          IF (DATA(LB).EQ.GRAPH(18)) GO TO 353
          IF (DATA(LB).EQ.GRAPH(6)) GO TO 354
          IF (DATA(LB).EQ.GRAPH(19)) GO TO 355
          IF (DATA(LB).EQ.GRAPH(14)) GO TO 356
          IF (DATA(LB).EQ.GRAPH(7)) GO TO 357
          IF (DATA(LB).EQ.GRAPH(13)) GO TO 357
      717 IF (DATA(LB).EQ.GRAPH(12)) GO TO 719
          IF (DATA(LB).EQ.GRAPH(27)) GO TO 359
      720 IF (DATA(LB).EQ.GRAPH(22)) GO TO 709
      721 LI = LB+1
          IF (DATA(LI).EQ.GRAPH(5)) GO TO 710
      722 IF (DATA(LB).EQ.GRAPH(20)) GO TO 711
          L = 9
          GO TO 300
      712 L = 14
          GO TO 400
      713 LSW(LC) = 06
          GO TO 1002
      351 LLD =LLC-1
          LSW(LC) = 40
          GO TO 1001
      352 LLD = LLC-1
          LSW(LC) = 41
          GO TO 1001
      353 IF (DATA(LA).NE.GRAPH(27)) GO TO 714
          LSW(LC) = 09
          GO TO 1002
      714 LI = LB+1
          IF (DATA(LI).NE.GRAPH(18)) GO TO 715
      716 LSW(LC) = 05
          GO TO 1002
      715 L = 15
          GO TO 400
      354 LI = LB+1
```

```
            IF (DATA(LI).NE.GRAPH(6)) GO TO 717
      718   LSW(LC) = 09
            GO TO 1002
      355   LI = LB+1
            IF (DATA(LI).EQ.GRAPH(19)) GO TO 718
            IF (DATA(LA).EQ.GRAPH(12)) GO TO 718
            IF (DATA(LA).EQ.GRAPH(18)) GO TO 718
            IF (DATA(LI).EQ.GRAPH(20)) GO TO 719
            [redacted]
            L = 16
[redacted]
            GO TO 400
      719   LSW(LC) = 40
            GO TO 1002
      356   IF (DATA(LA).EQ.GRAPH(4)) GO TO 718
            GO TO 717
      357   L = 17
            GO TO 400
      359   IF (DATA(LA).EQ.GRAPH(7)) GO TO 719
            IF (DATA(LA).EQ.GRAPH(8)) GO TO 719
            IF (DATA(LA).EQ.GRAPH(19)) GO TO 719
            IF (DATA(LA).NE.GRAPH(20)) GO TO 720
            LSW(LC) = 39
            GO TO 1002
      709   IF (DATA(LA).EQ.GRAPH(27)) GO TO 719
            GO TO 721
      710   LI = LB+2
            IF (DATA(LI).NE.GRAPH(27)) GO TO 722
            L = 18
            GO TO 300
      711   LI = LB+1
            IF (DATA(LI).EQ.GRAPH(20)) GO TO 713
            GO TO 712
C           C AND CK TEST
     3000   IF (DATA(I).NE.GRAPH(3)) GO TO 3010
            IF (DATA(LB).NE.GRAPH(11)) GO TO 1000
            LLD = LLC-1
            LSW(LC) = 12
            GO TO 1001
C           K TEST
     3010   IF (DATA(I).NE.GRAPH(11)) GO TO 3020
            IF (DATA(LB).NE.GRAPH(14)) GO TO 1000
            LSW(LC) = 50
            LC = LC+1
            LSW(LC) = 29
            GO TO 1001
C           D TEST
```

```
      3020 IF (DATA(I).NE.GRAPH(4)) GO TO 3030
           IF (DATA(LB).NE.GRAPH(4)) GO TO 1000
           LSW(LC) = 14
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
C          B TEST
      3030 IF (DATA(I).NE.GRAPH(2)) GO TO 3040
           IF (DATA(LB).NE.GRAPH(2)) GO TO 1000
           LSW(LC) = 13
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
C          G TEST
      3040 IF (DATA(I).NE.GRAPH(7)) GO TO 3050
           IF (DATA(LB).NE.GRAPH(7)) GO TO 1000
           LSW(LC) = 15
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
C          F TEST
      3050 IF (DATA(I).NE.GRAPH(6)) GO TO 3060
           IF (DATA(LB).NE.GRAPH(6)) GO TO 1000
           LSW(LC) = 18
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
C          L TEST
      3060 IF (DATA(I).NE.GRAPH(12)) GO TO 3070
           IF (DATA(LB).NE.GRAPH(12)) GO TO 1000
           LSW(LC) = 27
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
C          N TEST
      3070 IF (DATA(I).NE.GRAPH(14)) GO TO 3080
           IF (DATA(LB).EQ.GRAPH(11)) GO TO 360
           IF (DATA(LB).EQ.GRAPH(7)) GO TO 361
           IF (DATA(LB).NE.GRAPH(14)) GO TO 1000
           LSW(LC) = 29
           LC = LC+1
           LSW(LC) = 50
           GO TO 1001
       360 LSW(LC) = 30
           GO TO 1002
       361 LLD = LLC-1
           LSW(LC) = 30
           GO TO 1001
```

```
                C      P TEST
0365             3080 IF (DATA(I).NE.GRAPH(16)) GO TO 3090
0366                  IF (DATA(LB).NE.GRAPH(16)) GO TO 1000
0367                  LSW(LC) = 10
0368                  LC = LC+1
0369                  LSW(LC) = 50
0370                  GO TO 1001
                C      R TEST
0371             3090 IF (DATA(I).NE.GRAPH(18)) GO TO 1000
0372                  IF (DATA(LB).NE.GRAPH(18)) GO TO 1000
0373                  LSW(LC) = 32
0374                  LC = LC+1
0375                  LSW(LC) = 50
0376                  GO TO 1001
                C      PRECEEDING VOWEL TEST
0377              300 IF (DATA(LB).EQ.GRAPH(1)) GO TO 302
0378                  IF (DATA(LB).EQ.GRAPH(5)) GO TO 302
0379                  IF (DATA(LB).EQ.GRAPH(9)) GO TO 302
0380                  IF (DATA(LB).EQ.GRAPH(15)) GO TO 302
0381                  IF (DATA(LB).EQ.GRAPH(21)) GO TO 302
0382                  GO TO 301
                C      PRECEEDING VOWEL TEST
0383              400 IF (DATA(LA).EQ.GRAPH(1)) GO TO 302
0384                  IF (DATA(LA).EQ.GRAPH(5)) GO TO 302
0385                  IF (DATA(LA).EQ.GRAPH(9)) GO TO 302
0386                  IF (DATA(LA).EQ.GRAPH(15)) GO TO 302
0387                  IF (DATA(LA).EQ.GRAPH(21)) GO TO 302
                C      CONSONENT
0388              301 IF (L.EQ.1) GO TO 310
0389                  IF (L.EQ.2) GO TO 320
0390                  IF (L.EQ.3) GO TO 704
0391                  IF (L.EQ.4) GO TO 340
0392                  IF (L.EQ.5) GO TO 321
0393                  IF (L.EQ.6) GO TO 330
0394                  IF (L.EQ.7) GO TO 732
0395                  IF (L.EQ.8) GO TO 318
0396                  IF (L.EQ.9) GO TO 712
0397                  IF (L.EQ.13) GO TO 708
0398                  IF (L.EQ.14) GO TO 713
0399                  IF (L.EQ.15) GO TO 716
0400                  IF (L.EQ.16) GO TO 719
0401                  IF (L.EQ.17) GO TO 718
0402                  IF (L.EQ.18) GO TO 719
                C      VOWEL
0403              302 IF (L.EQ.1) GO TO 1000
0404                  IF (L.EQ.2) GO TO 323
0405                  IF (L.EQ.3) GO TO 703
0406                  IF (L.EQ.4) GO TO 341
```

```
              IF (L.EQ.5) GO TO 323
              IF (L.EQ.6) GO TO 333
              IF (L.EQ.7) GO TO 734
              IF (L.EQ.8) GO TO 317
              IF (L.EQ.9) GO TO 1000
              IF (L.EQ.13) GO TO 707
              IF (L.EQ.14) GO TO 1000
              IF (L.EQ.15) GO TO 717
              IF (L.EQ.16) GO TO 717
              IF (L.EQ.17) GO TO 717
              IF (L.EQ.18) GO TO 1000
         1000 CONTINUE
              IF(DATA(I).EQ.GRAPH(M)) GO TO 21
           11 CONTINUE
           21 LSW(LC) = LPHONE(LM)
              LC = LC+1
              IF (LJ.EQ.1) GO TO 501
              GO TO 12
          501 LSW(LC) = 50
              LC = LC+1
              LF = LB+1
           12 CONTINUE
              LLD = LLC+1
              LLC = LLC+1
              LSW(LLD) = 49
              WRITE(6,4) (DATA(I), I=LD,LLC)
            4 FORMAT(1H0, 15A1)
              WRITE(6,5) (LSW(I), I=LC,LLD)
            5 FORMAT(1X, 15I2)
              WRITE(6,6)
            6 FORMAT(1X, 1H )
              J = K
              JA = K
              JAA = 0
              LCOUNT = 1
              LJ = 0
              GO TO 100
         2000 CONTINUE
              STOP
              END
```

```
HASP-II JOB STATISTICS --    482 CARDS READ --      732 LINES PRINTED --

                                               H A S P   S Y S T E M

$ 14.14.04 JOB 143 -- SIMUREAD -- BEGINNING EXEC - INIT 3 - CLASS A
 *14.14.51 JOB 143 IHC0021 STOP      0
N 14.14.57 JOB 143 END EXECUTION.
//SIMUREAD      JOB (1378,C03),'LETON'                                        JOB
IEF236I ALLOC. FOR SIMUREAD COMPILE
IEF237I SYSPRINT ON 02C
IEF237I SYSPUNCH ON C29
IEF237I SYSLIN   ON 233
IEF237I SYSIN    ON C21
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000476     DELETED
IEF285I    VOL SER NOS=         .
IEF285I    SYS69084.T131453.RV000.SIMUREAD.LOADSET      PASSED
IEF285I    VOL SER NOS= UHSYS4.
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000478     DELETED
IEF285I    VOL SER NOS=         .
** STEP COMPILE   ENDED.  STEP CPU TIME =     9.69 SEC., STEP I/O COUNT =
IEF236I ALLOC. FOR SIMUREAD LOAD
IEF237I SYSLIB   ON 231
IEF237I          ON 235
IEF237I SYSPRINT ON 02C
IEF237I SYSLIN   ON 233
IEF237I SYSLMOD  ON 233
IEF237I SYSUT1   ON 235
IEF285I    SYS1.FORTLIB                                 KEPT
IEF285I    VOL SER NOS= UHSYS2.
IEF285I    SYS1.FORTSUB                                 KEPT
IEF285I    VOL SER NOS= UHSYS6.
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000479     DELETED
IEF285I    VOL SER NOS=         .
IEF285I    SYS69084.T131453.RV000.SIMUREAD.LOADSET      DELETED
IEF285I    VOL SER NOS= UHSYS4.
IEF285I    SYS69084.T131453.RV000.SIMUREAD.GOSET        PASSED
IEF285I    VOL SER NOS= UHSYS4.
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000480     DELETED
IEF285I    VOL SER NOS= UHSYS6.
** STEP LOAD      ENDED.  STEP CPU TIME =     0.72 SEC., STEP I/O COUNT =
IEF236I ALLOC. FOR SIMUREAD GO
IEF237I PGM=*.DD ON 233
IEF237I FT05F001 ON C22
IEF237I FT06F001 ON 02C
IEF237I FT07F001 ON 029
IEF285I    SYS69084.T131453.RV000.SIMUREAD.GOSET        PASSED
IEF285I    VOL SER NOS= UHSYS4.
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000483     DELETED
IEF285I    VOL SER NOS=         .
IEF285I    SYS69084.T131453.RV000.SIMUREAD.R0000481     DELETED
IEF285I    VOL SER NOS=         .
** STEP GO        ENDED.  STEP CPU TIME =     0.84 SEC., STEP I/O COUNT =
IEF285I    SYS69084.T131453.RV000.SIMUREAD.GOSET        DELETED
IEF285I    VOL SER NOS= UHSYS4.
** ENDED JOB #1378 - SIMUREAD BY LETON                 , CPU TIME =    11.3 SE
```

OUTPUT
PRE-PRIMER
CORPUS

PAGES 55-57

AM.
32849

NAN.
29 32949

AN.
32949

ANN.
3295049

MAN.
28 32949

MEN.
28 22949

IN.
12949

ON.
62949

RAN.
32 32949

RAM.
32 32849

RIM.
32 12849

RUN.
32 52949

A.
549

SUN.
23 52949

SAM.
23 32849

SUM.
23 52849

AS.
32549

US.
52549

IS.
12549

RUNS.
32 5292349

CAN.
14 32949

ADD.
3145049

CAM.
14 32849

MAD.
28 31449

DEN.
14 22949

SAD.
23 31449

DIN.
14 12949

DID.
14 11449

ED.
21449

SAD.
23 31449

NED.
29 21449

DIM.
14 12849

RED.
32 21449

RID.
32 11449

AND.
3291449

MEND.
28 2291449

END.
2291449

SAND.
23 3291449

SEND.
23 2291449

TED.
11 21449

TAN.
11 32949

TIM.
11 12849

TIN.
11 12949

TEN.
11 22949

TOT.
11 61149

IT.
11149

SET.
23 21149

SAP.
23 31049

POD.
10 61449

PAN.
10 32949

PAST.
10 3231149

PEN.
10 22949

PET.
10 21149

POT.
10 61149

RAP.
32 31049

STOP.
2311 61049

PIN.
10 12949

PIT.
10 11149

TOP.
11 61049

GAP.
15 31049

STEP.
2311 21049

PIG.
10 11549

SIP.
23 11049

RIP.
32 11049

TIP.
11 11049

PEG.
10 21549

UP.
51049

GRUNT.
1532 5291149

GRIP.
1532 11049

GRIN.
1532 12949

DRUG.
1432 51549

DRIP.
1432 11049

GRIT.
1532 11149

DRAG.
1432 31549

DROP.
1432 61049

SPAN.
2310 32949

GRASP.
1532 3231049

STAMP.
2311 3281049

STUMP.
2311 5281049

SPENT.
2310 2291149

SPIT.
2310 11149

SPIN.
2310 12949

SPOT.
2310 61149

CAT.
12 31149

CRISP.
1232 1231049

CAST.
12 3231149

CAN.
12 32949

CREPT.
1232 2101149

CAMP.
12 3281049

CAP.
12 31049

CUT.
12 51149

COT.
12 61149

CUP.
12 51049

ACT.
3121149

CROP.
1232 61049

COST.
1240231149

HAT.
20 31149

HOP.
20 61049

HIM.
20 12849

HAS.
20 32549

HOT.
20 61149

HID.
20 11449

HAT.
20 31149

HUM.
20 52849

HIP.
20 11049

HAM.
20 32849

HUT.
20 51149

HIS.
20 12549

HIT.
20 11149

HUG.
20 51549

HUNT.
20 5291149

HAND.
20 3291449

FAN.
18 32949

FUN.
18 52949

FAD.
18 31449

FIT.
18 11149

FIG.
18 11549

FOG.
18 91549

FAST.
18 3231149

FRED.
1832 21449

FRET.
1832 21149

FAT.
18 31149

FED.
18 21449

FIN.
18 12949

FIST.
18 1231149

FACT.
18 3121149

RAFT.
32 3181149

FROM.
1832 92849

FRONT.
1832 6291149

SIFT.
23 1181149

SHIFT.
1432 1181149

ON.
62949

MISS.
28 1235049

MESS.
28 2235049

MUSS.
26 5235049

DON.
14 92949

TOM.
11 92849

DOG.
14 91549

GRASS.
1532 3235049

EGG.
2155049

COG.
12 91549

CROSS.
1232 9235049

FOR.
18 53249

FROG.
1832 91549

OFF.
9185049

PHASE II OUTPUT, PRIMER CORPUS

Pages 58-104

ANT.
3291149

ANTLERS.
3291127 5322349

AFTER.
31811 53249

APPLE.
3105027549

AND.
3291449

ATE.
36115049

AT.
31149

ASK.
3231249

ALL.
3275049 3 → 9

ARAB.
632 31349 6 → 2

ARE.
6325049

ART.
6321149

AIMED.
3628 21449 214 → 50/14

AIM.
362849

DART.
14 6321149

SCARF.
2312 6321849

PART.
10 6321149

CART.
12 6321149

HARD.
20 6321449

HARM.
20 6322849

TART.
11 6321149

CARD.
12 6321449

DARN.
14 6322949

MAR.
28 63249

CAR.
12 63249

START.
2311 6321149

STARTER.
2311 63211 53249

FARM.
18 6322849

FARMER.
18 63228 53249

BAR.
13 63249

BARN.
13 0322949

BARK.
13 6321249

MARK.
28 0321249

DARK.
14 6321249

LARK.
27 0321249

PARK.
10 0321249

PAR.
10 03249

MAR.
28 63249

FAR.
18 03249

STAR.
2311 03249

BAR.
13 03249

JAR.
17 03249

WARM.
31 9322849

WASP.
31 6231049

SWAN.
2331 02949

SWAT.
2331 01149

WANT.
31 0291149

WATER.
31 011 53249

6 → 9

HAS.
20 32549

BASS.
13 3235649

LAST.
27 3231149

CLASS.
1227 3235649

GRASS.
1532 3235649

FAST.
18 3231149

FASTER.
18 32311 53249

MASK.
28 3231249

TASK.
11 3231249

PASS.
13 3235649

BLAD. input error
1327 31449

LAD.
27 31449

SAD.
23 31449

HAD.
20 31449

LADS.
27 3142349

23 → 25

BAG.
13 31449

SADDLE.
23 31453275549

PACK.
10 31249

RACK.
32 31249

TRACK.
1132 31249

TRACKS.
1132 3122349

TACK.
11 31249

JACK.
17 31249

SNACK.
2329 31249

SMACK.
2328 31249

STACK.
2311 31249

SACK.
23 31249

LACK.
27 31249

BLACK.
1327 31249

BACK.
13 31249

RAFT.
32 3161149

RAT.
32 31149

FLAT.
1827 31149

FAT.
18 31149

CATS.
12 31123549

SAT.
23 31149

SLAT.
2327 31149

MAT.
28 31149

BAT.
13 31149

CAT.
12 31149

RATTLE.
32 3115027549

CAMP.
12 3281049

CAMPER.
12 32810 53249

CAMEL.
12 328 22749

JAM.
17 32849

JAMES.
17 328 22549 3 → 36 2 → 50

DAMP.
14 3281049

DAMPER.
14 32810 53249

SAM.
23 32849

CLAM.
1227 32849

SLAM.
2327 32849

HAND.
20 3291449

BAN.
13 32949

CLAN.
1227 32949

VAN.
21 32949

CAN.
12 32949

JAN.
17 32949

DRANK.
1432 3301249

DAB.
14 31349

JAB.
17 31349

CRAB.
1232 31349

CAB.
12 31349

SLAB.
2327 31349

STAB.
2311 31349

PAN.
10 32949

PLANT.
1027 3291149

PLAN.
1027 32949

RANK.
32 3301249

TANK.
11 3301249

TAN.
11 32949

MAN.
28 32949

FRANK.
1832 3301249

TAB.
11 31349

RAGS.
37 3152349

23 → 25

FLAG.
1827 31549

JAG.
17 31549

LAG.
27 31549

BRAG.
1332 31549

BAG.
13 31549

WAG.
31 31549

LAP.
27 31049

SLAP.
2327 31049

CLAP.
1227 31049

CAP.
12 31049

CANAL.
12 329 32749

HAVE.
2036215 49

36 → 3

AGAIN.
315362949

BAIL.
13362749

BRAID.
13323601449

BAIT.
13361149

BRAIDED.
13323614 11449

1 ⇒ 5

BRAIN.
13323062949

CLAIM.
12273628 49

BRAIN.
1332362949

FAIL.
18362749

FAINT.
1836291149

GAIL.
15362749

GAIN.
15362949

GAIT.
15361149

GRAIN.
1532362949

RAIL.
20362749

HAIR.
20 23249

JAIL.
17362749

LAIR.
27 23249

MAID.
28361449

MAIL.
28362749

MAIN.
28362949

NAIL.
29362749

PAID.
10361449

PAIN.
10362949

PAINT.
1036291149

PAIR.
10 23249

PLAIN.
1027362949

PRAISE.
103236235049
5

23 → 25

RAIL.
32362749

RAIN.
32362949

RAISE.
3236235649 23 → 25

RAISED.
323623 21449

SAID.
23361449 36 → 2

SAIL.
23362749

SAILOR.
233627 53249

SLAIN.
2327362949

SPAIN.
2319362949

SPFIAN. input error
231032 1 32949

STAIN.
2311362949

STRAIN.
231132362949

RAISED.
323623 21449

SAID.
23361449

SAIL.
23362749

SAILOR.
233627 53249

SLAIN.
2327362949

repeat of above

SPAIN.
2310362949

repeat

SPRAIN.
231032362949

Correction

STAIN.
2311362949

repeat

STRAIN.
231132362949

TAIL.
11362749

TRAILER.
11323627 53249

TRAIN.
1132362949

VACANT.
21 312 3291149

3 → 36

WAIST.
3136231149

SCARED.
2312 632 21449

6 → 2

214 → 14

BEAD.
13351449

BEAK.
13351249

BEAM.
13352849

BEAN.
13352949

BEAST.
1335231149

BEARD.
13 1321449

BEAT.
13351149

DEAR.
14 13249

EAR.
13249

EASE.
35235.49

23 → 25

EAST.
35231149

EAT.
351149

EAVE.
35215.49

FEAR.
16 13249

FEAST.
1835231149

HEAL.
20352749

HEAT.
20351149

JEAN.
17352949

LEAD.
27351449

LEAF.
27351649

LEAK.
27351249

LEAN.
27352949

LEAP.
27351349

LEASE.
2735235049

LEAST.
2735231149

LEAVE.
2735215049

MEAL.
28352749

MEAN.
28352949

MEAT.
28351149

NEAR.
29 13249

NEAT.
29351149

PEA.
103549

PEAK.
10351249

PLEASE.
102735235049 23→25

PLEAT.
1027351149

SEA.
233549

SEAL.
23352749

SEAM.
23352849

STEAM.
2311352849

READ.
32351449

REAP.
32351?49

REAR.
32 13249

SEAT.
23351149

TEA.
113549

TEAM.
11352849

TEASE.
1135235049 23 → 25

WEAVE.
3135215049

BALD.
13 3271449 3 → 9

WALL.
31 3275049 3 → 9

CALLED.
12 32750 21449 3 → 9 214 → 14

SMALL.
2328 3275049 3 → 9

STALL.
2311 3275049 3 → 9

FALL.
18 3275049 3 → 9

TALL.
11 3275049 3 → 9

HALL.
20 3275049 3 → 9

CALL.
12 3275049 3 → 9

BALL.
13 3275049 3 → 9

CAME.
1236285049

FADE.
1836145049

DATE.
1436115049

DARE.
14 6325049 6 → 2

WAVE.
3136215049

WARE.
31 9325049 9 → 2

WAKE.
3136125049

TRADE.
113236145049

TASTE.
11 323115049 3 → 36

TAME.
1136285049

TALE.
1136275049

TAKE.
1136125049

STATE.
231136115049

STARE.
2311 6325049 6 → 2

SNAKE.
232936125049

STALE.
23113627549

SLATE.
232736115049

SKATE.
231236115049

SAVE.
2336215049

SAME.
2336285049

SALE.
2336275049

SAKE.
2336125049

SAFE.
2336185049

RATE.
3236115049

RAKE.
3236125049

PLANE.
102736295049

PAVE.
1036215049

PAKE.
10 6325049   6 → 2

PANE.
1036295049

PALE.
1036275049

NAME.
2936265049

MAKE.
28 6325049   6 → 2

MATE.
2836115049

MANE.
2836295049

MALE.
2836275049

MAKE.
2836125049

MADE.
2836145049

LATE.
2736115049

LANE.
2736295049

LAKE.
2736125049

KATE.
1236115049

HATE.
2036115049

HARE.
20 6325049 6 → 2

GAVE.
1536215049

GATE.
1536115049

GALE.
1536275049

GAME.
1536285049

FLARE.
1827 6325049 6 → 2

FATE.
1836115049

FAKE.
18 6325049 6 → 2

BAKE.
1336125049

BLAME.
132736285049

BASE.
1336235049

BALE.
1336275049

BARE.
13 6325049 6 → 2

CANE.
1236295049

CAPE.
1236105049

CAVE.
1236215049

CARE.
12 6325049 6 → 2

PULLET.
10 72750 21149

PUTS.
10 5112349 5 → 7

BULL.
13 7275049

FULL.
18 7275049

PULL.
10 7275049

UPON.
510 62949

UNDER.
52914 53249

DUG.
14 51549

TUG.
11 51549

BUG.
13 51549

PLUG.
1027 51549

BUT.
13 51149

BUCKET.
13 512 21149

DUCKS.
14 5122349

DUCK.
14 51249

TRUCK.
1132 51249

STUCK.
2311 51249

PLUCK.
1027 51249

BUCK.
13 51249

TUCK.
11 51249

LUCK.
27 51249

JUMP.
17 5261549

HUMP.
20 5281049

LUMBER.
27 52813 53249

BUMP.
13 5281C49

JUNK.
17 5301249

GUN.
15 52949

RUN.
32 52949

RUNS.
32 5292349 23 → 25

HUNT.
20 5291149

HUNTER.
20 52911 53249

HUNTED.
20 52911 11449 1 → 5 P

BUN.
13 52949

SUNK.
23 5301249

RUBS.
32 5132349 23 → 25

STUB.
2311 51349

RUB.
32 51349

TUB.
11 51349

CUB.
12 51349

HUB.
20 51349

JUST.
17 5231149

PUDDLE.
10 51450275C49

MUD.
28 51449

BULB.
13 5271549

SWARM.
2331 9322849

WAR.
31 93249

WART.
31 9321149

JAW.
17 949

CAW.
12 949

DAWN.
14 92949

FAWN.
18 92949

RAW.
32 949

PAW.
10 949

SAW.
23 949

DRAW.
1432 949

VACANT.
21 312 3291149

3 → 6

CANAL.
12 329 32749

JOB.
17 61349

COB.
12 61349

ROB.
32 61349

BLACKS.
1327 3122349

STOCK.
2311 61249

LOCK.
27 61249

MUCK.
28 61249

ROCK.
32 61249

SOCK.
23 61249

DOCK.
14 61249

JOT.
17 61149

GOT.
15 61149

HOP.
20 61049

TOP.
11 61049

STOP.
2311 61049

MONSTER.
28 6292311 53249

TOPS.
11 6102349

BOND.
13 6291449

NOW.
294149

VOW.
214149

~~S~~OWN.
23412949

down
1441 2949

TOWN.
11412949

HOW.
204149

COW.
124149

OAT.
401149

ROAM.
32402849

CLOAK.
1227401249

BOARD.
1340321449

40 → 9

BOAT.
13401149

GOAL.
15402749

OAK.
401249

LOAF.
27401649

SOAP.
23401049

HOARSE.
20403223550649

40 → 9

SOAR.
23403249

40 → 9

MOAN.
28402949

OAR.
403249

40 → 9

ROAD.
32401449

COAT.
12401149

COARSE.
12403223550649

40 → 9

ROAR.
32403249

40 → 9

GROAN.
1532402949

GOAT.
15401149

TOAD.
11401449

CROAK.
1232401249

FOAM.
18402849

SOAK.
23401249

COAL.
12402749

JOAN.
17402949

ROAST.
3240231149

CORRAL.
12 53250 32749 5 → 9

FROM.
1832 92849 9 → 5

WOLF.
3140271649 40 → 7

COME.
12 9265049 9 → 5

DONE.
14 9295049 9 → 5

SOME.
23 9265049 9 → 5

WAGON.
31 315 62949

DOES.
14402549 40 → 5

GLOVE.
1527402150049 40 → 5

DOVE.
1440215049

LOVE.
2740215049 40 → 5

FORD.
18 5321449 5 → 9

CORK.
12 5321249 5 → 9

FORK.
18 5321249 5 → 9

FOR.
18 53249 5 → 9

OR.
93249

NOR.
29 53249 5 → 9

ALONE.
32740295049 3 → 5

JONES.
17 029 22549 6 → 40 225 → 25

LONE.
2740295049

BONE.
1340295049

CONE.
1240295049

STONE.
231140295049

DRONE.
143240295049

TONE.
1140295049

HOPE.
2040105049

ROPE.
3240105049

LOPE.
2740105049

WORE.
31 5325049 5 → 9

FORE.
18 5325049 5 → 9

TORE.
11 5325049 5 → 9

LUKE.
27 5325049 5 → 9

TURN.
11 5322949 5 → 9

WOKE.
31 5325049 5 → 7

BEFORE.
13 216 5325049 2 → 35 5 → 9

POKE.
10 5325049 5 → 9

OKE.
9525049

COKE.
12 5325049 5 → 9

BORE.
13 5325049 5 → 9

SNORE.
2329 5325049 5 → 9

MORE.
28 5325049 5 → 9

SURE.
23 5325049 5 → 9

VOTE.
2140115049

NOTE.
2940115049

JOKE.
1740125049

SMOKE.
232840125049

POKE.
1040125049

POLE.
1040275049

SOLE.
2340275049

MOLE.
2840275049

HOLE.
2040275049

STOLE.
231140275049

DOSE.
1440235049

✓NOSE.
2940235049 23 → 25

✓ROSE.
32 9235049 9 → 40 23 → 25

GLOBE.
152740135049

ROBE.
3240135049

RODE.
3240145049

STOVE.
231140215049

COVE.
1240215049

WOVE.
3140215049

✓ROVER.
32 621 53249 6 → 40

✓HOME.
20 9285049 9 → 40

✓DOME.
14 9285049 9 → 40

TOLD.
1140271449

HOLD.
2040271449

OLD.
40271449

OVER.
4021 53249

BOLT.
1340271149

COLT.
1240271149

VOLT.
2140271149

REVOLT.
32 2214027l149 2 ⇒ 35

HOLSTER.
2040272311 53249

RULLS.
3240275023349 23 ⇒ 25

POST.
1040231149

HOST.
2040231149

MOST.
2840231149

HOSTESS.
20402311 2233349

GO.
154049

NO.
29 049 6 ⇒ 40

SO.
234049

CORN. 5 → 9
12 5322949

LORD. 5 → 9
27 5321449

BORN. 5 → 9
13 5322949

ACORN. 3 → 36; 5 → 9
312 5322949

FOREST. 5 → 9
18 532 2231149

CORD. 5 → 9
12 5321449

HORSE. 5 → 9
20 5322353449

HORN. 5 → 9
20 5322949

WORN. 5 → 9
31 5322949

FROG. 9 → 6
1832 91549

BOG. 9 → 6
13 91549

ACROSS. 3 → 5
31232 9235349

LOST.
27 9231149

LOG. 9 → 6
27 91549

TOM. 9 → 6
11 92849

DON. 9 → 6
14 92949

JOG.
17 91549

9 ⇒ 6

TOSSED.
11 92350 21449

214 ⇒ 11

OFF.
9185049

FLOCK.
1827 61249

BOB.
13 61349

SLOT.
2327 61149

BLOT.
1327 61149

PLOT.
1027 61149

CLOD.
1227 61449

BOTTLE.
13 611502750 49

FLOP.
1827 61049

SLOP.
2327 61049

MOVE.
2840215049

40 ⇒ 39

TO.
113949

WOE.
314049

TOE.
114049

FOE.
184049

HOE.
204049

GOES.
15402549

HIDE.
2037145049

WIDE.
3137145049

BRIDE.
133237145049

STRIDE.
23113237145049

SIDE.
2337145049

RIDE.
3237145049

LIFE.
2737185049

WIFE.
3137185049

LIKE.
2737125049

DIKE.
1437125049

PIKE.
1037125049

SPIKE.
231037125049

STRIKE.
23113237125049

STILE.
231137275049

SMILE.
232837275049

MILE.
2837275049

TILE.
1137275049

PILE.
1037275049

VILE.
2137275049

NINE.
2937295049

FINE.
1837295049

LINE.
2737295049

SPINE.
231037295049

PINE.
1037295049

SWINE.
233137295049

TIME.
1137285049

LIME.
2737285049

CRIME.
123237285049

DIME.
1437285049

PIPE.
1037105049

RIPE.
3237105049

TIRE.
1137325049

HIKE.
2037325049

SPIRE.
2310373250 49

FIRE.
1837325049

SIRE.
2337325049

SPITE.
2310371150 49

KITE.
1237115049

BITE.
1337115049

LIVE.
2737215049

DIVER.
143721 53249

FIVE.
1837215049

HIVE.
2037215049

DIVE.
1437215049

DINNER.
14 12950 53249

FIND.
1837291449

FINER.
183729 53249

KIND.
1237291449

MIND.
2837291449

BIND.
1337291449

GRIND.
153237291449

BEHIND.
13 22037291449 2 → 35

FIRED.
18 532 21449

SIR.
23 53249

FIR.
18 53249

FIRM.
18 1322849 1 → 5

BIRD.
13 5321449

GIRL.
15 1322749 1 → 5

DIRT.
14 1321149 1 → 5

FIRST.
18 532231149

FERN.
18 5322949

JERK.
17 5321249

NEED.
29351449

STEEP.
2311351049

WEEK.
31351249

KEEN.
12352949

DEEPER.
143510 53249

FEELER.
183527 53249

KEEPER.
123510 53249

FIFTEEN.
18 11811352949

STEED.
2311351449

MEET.
28351149

DEER.
14353249

REEL.
32352749

DEEP.
14351049

KEEP.
12351049

PEEK.
10351249

PEEKED.
103512 21449 214 → 11

PEEL.
10352749

FEEL.
18352749

TREE.
11323549

JEEP.
17351049

SLEEK.
2327351249

GREEN.
1532352949

FEED.
18351449

FEET.
18351149

SEEM.
23352849

SEE.
233549

SEED.
23351449

SEEK.
23351249

STEEL.
2311352749

FLEE.
18273549

BEET.
13351149

BEEF.
13351849

HE.
203549

WE.
313549

PETE.
1035115049

HERE.
2035325049 35 → 1

BE.
133549

BEEN.
13352949 35 → 1

WET.
31 21149

LESS.
27 2235049

LET.
27 21149

LEG.
27 21549

HELMET.
20 22728 21149

BEND.
13 2291449

WELL.
31 2275049

WENT.
31 2291149

WEST.
31 2231149

WESTERN.
31 22311 5322949

TEND.
11 2291449

TENDER.
11 22914 53249

PEPPER.
10 21050 53249

TENNIS.
11 22950 12549

DENTED.
14 22911 11449

DENT.
14 2291149

MENDED.
28 22914 11449

MENDER.
28 22914 53249

MEND.
28 2291449

BEN.
13 22949

BEST.
13 2231149

BED.
13 21449

BEG.
13 21549

BELT.
13 2271149

BLED.
1327 21449

BET.
13 21149

PECK.
10 21249

DECK.
14 21249

NECK.
29 21249

KEG.
12 21549

KEPT.
12 2101149

GET.
15 21149

JESS.
17 2235049

ELLEN.
22750 22949

EGGS.
2155022349

EGG.
2155049

WETTER.
31 21150 53249

TED.
11 11449

1 → 2

TEN.
11 22949

HELD.
20 2271449

SLED.
2327 21449

RED.
32 21449

ELF.
2271849

PRESS.
1032 2235049

SENT.
23 2291149

BELT.
13 2271149

STEPS.
2311 2102349

HEN.
20 22949

DRESS.
1432 2235049

TENT.
11 2291149

JET.
17 21149

JEFF.
17 2185049

VENT.
21 2291149

VEST.
21 2231149

PET.
10 21149

BEGS.
13 2152349 23 → 25

NELL.
29 2275049

ALIVE.
32737215049 3 → 5

TIE.
113749

LIE.
273749

DIE.
143749

PIE.
103749

CRIED.
1232371449

CRIED.
1432371449

TRIED.
1132371449

HIS.
20 12549

IS.
12549

BIB.
13 11349

RIB.
32 11349

BLIMP.
1327 1281049

BRIM.
1332 12849

JIM.
17 12849

LICK.
27 11249

WICK.
31 11249

PICK.
10 11249

TICK.
11 11249

SICK.
23 11249

STICK.
2311 11249

NICK.
29 11249

TRICK.
1132 11249

KICK.
12 11249

BRICKS.
1332 1122349

BRICK.
1332 11249

LINT.
27 1291149

WINTER.
31 12911 53249

WIN.
31 12949

BIN.
13 12949

PINK.
10 1301249

INK.
1301249

SINK.
23 1301249

MINK.
28 1301249

RINK.
32 1301249

DRINK.
1432 1301249

WINK.
31 1301249

LINK.
27 1301249

KIN.
12 12949

MILL.
28 1275049

STILL.
2311 1275049

TILL.
11 1275049

HILL.
20 1275049

SILL.
23 1275049

WILL.
31 1275049

PILL.
10 1275049

MILK.
28 1271249

SILK.
23 1271249

KILL.
12 1275049

KILT.
12 1271149

SPILLED.
2310 12750 21449   214 → 14

JILL.
17 1275049

SILVER.
23 12721 53249

LIST.
27 1231149

KISS.
12 1235049

RISK.
32 1231249

BRISK.
1332 1231249

NIP.
29 11049

LIP.
27 11049

FLIP.
1827 11049

SLIP.
2327 11049

TRIP.
1132 11049

SKIPPED.
2312 11050 21449   2/4 → 11

SITS.
23 1112349

SLIT.
2327 11149

BIT.
13 11149

KIT.
12 11149

LITTLE.
27 11150275049

KITTEN.
12 11150 22949

IT.
11149

HIT.
20 11149

WIG.
31 11549

BIG.
13 11549

DIG.
14 11549

JIG.
17 11549

TWIG.
1131 11549

KID.
12 11449

DID.
14 11449

HID.
20 11449

LIVE.
2737215049

SLIVER.
23273721 53249 37 → 1

GIVE.
1537215049 37 → 1

LIVER.
273721 53249 37 → 1

INTO.
129113349

HASP-II JOB STATISTICS -- 536 CARDS READ -- 2,849 LINES PRINTED --

126/770

Table B

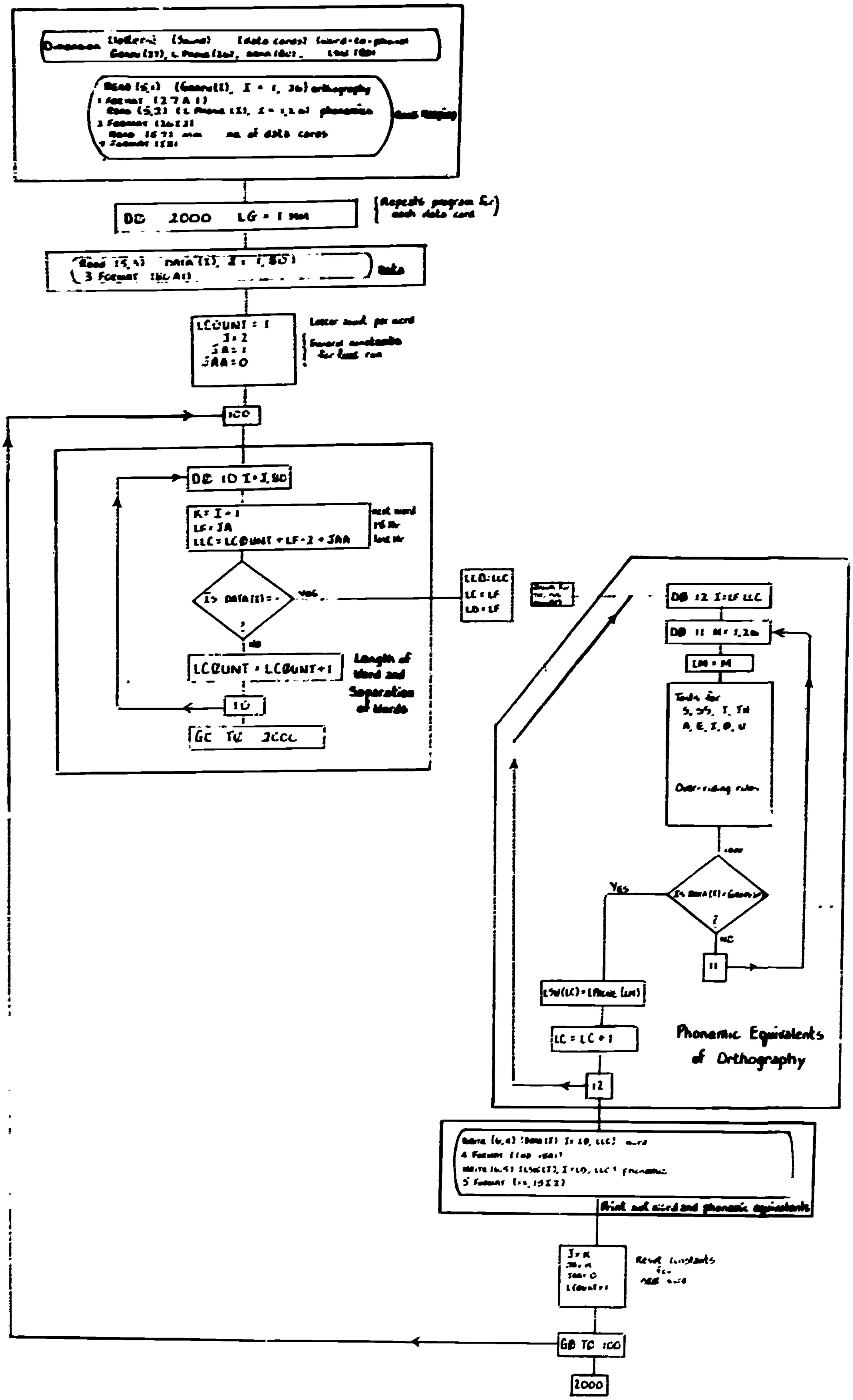
Pre-Primer and Primer Rules

Grapheme	Phoneme	Rule	Special Case
ai	02	/C__r	again said
ai	36	[Else]	
au	09	/C__¯V	
a	02	/C__re[# d]	
a	09	/C__lC /w__r¯V	water
a	05	/#__#	was
a	03		camel have
a	06	/¯V__r¯V /w__C-V	
a	36	/¯V__Ce /C__ste	vacant acorn
a	03	[Else]	
b	13		
c	12		
ck	12		
dd	14 50		
d	14		
ea	01	//[d # n f r] __r	beard
ea	02		bear
ea	35	/C__¯V	
ee	01	/C__r	been
ee	35	[Else]	
ed	12 14	/[d t] __#	
ed	11	/[ss k p] __#	

Grapheme	Phoneme	Rule	Special Case
ed	14	/[c v]__# [Else]	
e	01		here
e	05	/C__r¯V	roses
e	35	/#C__# /¯V__Ce#	
e	50	/ VC(c)__(s)#	
e	02	[Else]	
f	18		
gg	15 50		
g	15		
h	20		
ie	37	/¯VC__[# d]	
i	01		live sliver give liver
i	37	/C__Ce([d r])# /C__nd#	
i	05	/C__r¯V	
i	01	[Else] /¯V__¯V	
j	17		
k	12		
le	05 27	NC(C)__#	
ll	27 50		
l	27		
mm	28 50		
m	28		
n	30	/__k	

Grapheme	Phoneme	Rule	Special Case
nn →	29 50		
n →	29		
oa →	09	/¯V__r	
oa →	40	/¯V__C	
oa →	05		does
oe →	40	/C__(s)#	
ow →	41	//[n v h c]__# //[d t br]__n#	all cases in Primer & Pre-Primer
o →	05	/VC(C)__r#	from done glove love some dove come
o →	09	/¯V__[r ss ff ng]	
o →	07		wolf
o →	39		move to
o →	40	/¯V__Ce# /(C)C__# /C__lC /C__st	
o →	06	/C__C	
pp →	10 50		
p →	10		
rr →	32 50		
r →	32		
ss →	23 50		
s →	23	/#__ /__[k p t] //[k t p]__#	lease base coarse hoarse dose horse tennis

Grapheme	Phoneme	Rule	Special Case
s →	25	/V__V //[d m g n b n] (e)__# /V__#	
tt →	11 50		
t →	11		
u →	07	/C__11	
u →	05	/¯V__C¯V	
v →	21		
w →	31		
y →	33	/#__V	



Presentation Outline of

BASIC READING Skills

as Developed from Readiness through Grade 8

By

Glenn McCracken

Charles Walcutt

J. B. Lippincott Company
Philadelphia, Pa.

Readiness for Learning (Pierce McLeod)

A. Level One--Bilateral Training

Objectives:

1. Coordination of large-small muscles
2. Awareness of sensory stimuli produced by muscle activity
3. Mental image of body

B. Level Two--Unilateral Training

Objective:

Development and use of a predominant hand, foot, and eye

Skills for levels one and two developed through:

1. Motor training activities
2. Listening activities
3. Visual-motor training activities
4. Visual training activities

C. Level Three--Letter and Word Knowledge

Objectives:

1. Eye-hand coordination
2. Perceptual constancy
3. Perception of position in space
4. Perception of spatial relationships
5. Figure-ground perception (relating to letter configurations in word forms)

Skills developed through:

1. Visual-motor activities

   Tracing forms
   Completing figures
   Reproducing designs

2. Letter knowledge (Upper and lower case)

   Letter matching
   Letter completion
   Alphabet tracking
   Tachistoscopic training

3. Word study

   Word matching
   Word completion
   Tachistoscopic training

Linguistics and Reading

I. Culture

   A. Ideas
   B. Arts
   C. Facts
   D. Artifacts

II. Spoken Language

   A. Encoder--speaker
   B. Decoder--listener

III. Written Language

   A. Picture writing
   B. Word writing
   C. Alphabetic writing

      1. Phoneme--A unit sound of speech
      2. Grapheme--Significant unit of visual shape
      3. Morpheme--A word
      4. Encoder--writer
      5. Decoder--reader

IV. Stages in Reading Development

   A. Transfer--decoding stage
   B. Productive--meaning stage

      1. Lexical (dictionary meaning)
      2. Meaning conveyed by grammatical structure
      3. Meaning conveyed by social cultural reference

   C. Vivid imaginative realization stage

      1. Ideas from literature
      2. Scientific material

Pre-Primer

Recognition of colors
Left-to-right eye movement
Recognition of similarities and differences
Relationships
Concepts of more or less
Following sequence of events to tell a story
Awareness of page numbers
Counting
Recognition of position relationships

Oral expression
Recalling details to relate a story
Awareness of differences in shapes
Line-to-line sequence
Interpretation of a picture story
Making inferences
Understanding of ordinals (1-6)
Auditory discrimination
Associating and writing upper and lower case letters

Short vowel sounds
Consonants (m n r s d t p c g h f)
Blends (nd sp nt gr dr mp)
Capitals, period, comma, question mark
Combine sounds to form words, sentences, stories and poems

Primer

Long sounds of vowels

Vowels modified by (r w l ll ff)

Vowel digraphs

Final sound of (ed er le ck nk)

Consonants (b w l k j v)

Two-syllable and rhyming words
Paragraph, quotation marks, exclamation point, hyphen, apostrophe

Guided silent reading--comprehension

Prefix, suffix, root words

Compound words, singular, plural

BOOK 1-1

Sequence	Blends (sh ch tch th wh qu)
Colon, semicolon	Endings (ng ing y ay ey)
Abbreviations	Vowel sounds (oo - long and short)
Consonants (x y z soft c and g)	Dipththongs (ow ou oi oy)

BOOK 1-2

Vocabulary expansion	Parentheses
Classification of words	Antonyms
Contractions	Synonyms
Alphabetizing (number and first letter)	Homonyms
	Pronouns

GRADE ONE:

Vocabulary. . . . .2,197 words

Sound Spellings . .44 basic and variants

BOOK 2-1

Syllables (determining number)

Subject (Who or what did something?)

Verb (What happened?)

Nouns (Name words)

Adjectives (Descriptive words)

Adverbs

Direct object

Alphabetizing (second letter)

Review of sound spellings

Logical conclusions

Generalizations

BOOK 2-2

Syllabication (two-syllable words)

Telling a story from an outline

Preposition

Object of preposition

Verb (tense)

Predicate

Correct usage

Basic parts of a letter

Creative writing

BOOK 3-1

Skimming to find answers

Finding main idea in a paragraph

Reading to determine difference between fact and opinion

Reading for details

Finding evidence to support judgment

Predicting outcomes

Syllabication (three-syllable words)

Drawing conclusions

Interpreting mood and character

Figurative language

Comparatives and superlatives

Summarizing

Introduction to accent

BOOK 3-2

Outlining (simple)

Predicate noun

Predicate adjective

Possessive pronoun

Prepositional phrase

Positive and negative contractions

Syllabication (four-syllable words)

Interpreting humor

Reading to determine difference between fact and opinion

Reading to discover irrelevancy

Similes

Punctuation marks (all)

BOOK 4

Reading to follow directions
Review of sound spellings
Sentence structure
Index
Map skills
Interpreting punctuation marks
Dictionary exercises (guide words)
Outlining for details
Glossary
Indirect object
Participles
Dramatization
Interpreting the main idea
Making inferences--judgments

BOOK 5

Oral reading (interpretative)
Making inferences--character analysis
Compound and complex sentences
Figures of speech
Using reference material
Paragraph writing
Infinitives
Review sound spellings
Phrases (participial, adjectival, adverbial)
Clauses (subordinate, noun, adverbial)
Verbals--gerunds
Reading to determine between fact and opinion

BOOK 6

Word derivations
Interpretation of metaphors
Verb phrases
Analyzing structure and style
Perceiving relationships
Analogies
Colloquial usage and dialect
Classes of words
Basic sentence patterns:
- Noun-verb
- Noun-verb-noun
- Noun-linking verb-adjective
- Noun-linking verb-noun

Transitive and intransitive verbs

BOOKS 7 and 8

Continued vocabulary expansion

Basic sentence pattern work

Intonation: stress, pitch, juncture

Adjectives: base, derived

Clauses

Classification of nouns

Punctuation

Verbals--gerunds--infinitives

Data prepared by Miss Ruth E. Gorman, Reading Consultant for J. B. Lippincott Company.